Adaptation du contenu spatio-temporel des images pour un codage par ondelettes

Benjamin Le Guen

N° d'ordre: 3584

THÈSE

Présentée devant

devant l'Université de Rennes 1

pour obtenir

le grade de : Docteur de l'Université de Rennes 1
Mention Traitement du Signal et Télécommunications

par

Benjamin Le Guen

Équipe d'accueil : TECH/IRIS/CVA (Orange Labs, Rennes)
École Doctorale : Matisse
Composante universitaire : Supélec-SCEE/IETR-AC

Titre de la thèse :

Adaptation du contenu spatio-temporel des images pour un codage par ondelettes

soutenue le 14 février 2008 devant la commission d'examen

M.	Président	du jury	Président
M.	Michel	Barlaud	Rapporteurs
Mme	Béatrice	Pesquet-Popescu	
MM.	Kadi	Bouatouch	Examinateurs
	Vincent	Ricordel	
	Jacques	Palicot	
	Jacques	Weiss	
	Stéphane	Pateux	

Remerciements

Ce travail a été réalisé dans le laboratoire TECH/IRIS de France Télécom R&D au sein de l'équipe Compression Vidéo Avancée. Il est issu d'une collaboration avec l'équipe Supélec-SCEE, composante du laboratoire IETR.

Je voudrais tout d'abord remercier Vincent Marcatté et Alexandre Nolle pour m'avoir admis au sein du laboratoire IRIS. Je remercie sincèrement Henri Sanson et Ludovic Noblet pour m'avoir accueilli dans l'équipe CVA et m'avoir fourni les moyens de conduire et mettre en valeur mes recherches.

Un grand merci à Stéphane Pateux pour m'avoir guidé pendant ces trois années de thèse. Par son intuition, ses connaissances et son sens inné des mathématiques, Stéphane m'a permis d'avancer constamment dans mes recherches et de mener à bien les idées qui sont développées dans ce manuscrit.

Un grand merci également à Nathalie Cammas pour l'intérêt qu'elle a porté à ma thèse au quotidien. Sa grande disponibilité pour répondre à mes questions scientifiques et son soutien moral ont largement contribué à la réussite de cette thèse.

Je tiens à remercier sincèrement Jacques Palicot et Jacques Weiss, professeurs à Supélec, pour avoir accepté respectivement de diriger et d'encadrer cette thèse. Merci d'avoir pris le temps d'évaluer régulièrement la pertinence de ces travaux et de m'avoir encouragé à les mettre en valeur au travers de présentations et d'articles.

Je remercie M. Michel Barlaud, professeur à l'Université de Nice-Sophia Antipolis, et Mme Béatrice Pesquet-Popescu, professeur à l'ENST, d'avoir accepté la tâche difficile de rapporteurs. Je remercie M. Vincent Ricordel, maître de conférences à l'Université de Nantes, d'avoir accepter de juger ce travail. Enfin, un grand merci à M. Kadi Bouatouch, professeur à l'IFSIC, qui m'a fait l'honneur de présider ce jury.

Ce séjour à France Télécom R&D fut l'occasion de travailler en équipe et de rencontrer des personnes très sympathiques. A ce titre, je voudrais remercier Isabelle Amonou, Sylvain Kervadec et Maryline Clare pour leurs conseils, leur gentillesse et leur bonne humeur. Merci à Sid-Ahmed Berrani pour les nombreuses discussions que nous avons eues. Merci à mes compagnons du midi et à la joyeuse bande des thésards.

Merci à mes amis de Supélec et Georgia Tech.

Enfin, merci maman.

Sans l'aide et le soutien de tous, ces travaux de thèse n'auraient jamais pu aboutir.

Table des matières

Abréviations

SVH : Système Visuel Humain.
PSNR : Rapport Signal à Bruit.
SSIM : Structural Similarity Image Metric.
EQM : Erreur Quadratique Moyenne.

ITU : International Telecommunication Union.
ITU-T : ITU Telecommunication Standardization Sector.
ISO : International Organization for Standardization.
IEC : International Electrotechnical Commission.
JPEG : Joint Picture Expert Group.
MPEG : Motion Picture Expert Group.

AS t : Schéma par analyse-synthèse temporelles.
AS2D : Schéma par analyse-synthèse spatiales.
AS2D+t : Schéma par analyse-synthèse spatio-temporelles.
JPEG2000 : Norme actuelle de compression scalable d'images fixes.
H.264/MPEG-4 SVC : Norme actuelle de compression scalable de vidéos.

OBMC : Ovelapped Block Motion Compensation.
SOBMC : Switched Ovelapped Block Motion Compensation.
CGI : Control Grid Interpolation.
SCGI : Switched Control Grid Interpolation.

EZW : Embedded Zerotree Wavelet.
EBCOT : Embedded Block Coding with Optimized Truncation.
SPIHT : Set Partitioning in Hierarchical Tree.

GOF : Groupe d'images.
CIF : Common Intermediate Format 352×288.
DID : Différence d'image déplacée.

Notations

Généralités

$\mathcal{C}^\alpha(\mathbb{D})$: Ensemble des fonctions définies sur $\mathbb{D}$ α fois continues et dérivables.
$\mathcal{C}^\alpha \backslash \mathcal{C}^\alpha(\mathbb{D})$: Ensemble des fonctions définies sur $\mathbb{D}$ contenant des régions de régularité $\mathcal{C}^\alpha$ séparées par des singularités $\mathcal{C}^\alpha$.
$\tilde{f}_M$: Approximation non linéaire de f avec M coefficients.

$< ., . >$: produit scalaire.
$\|.\|^2$: norme L_2.

Γ : flux géométrique.
γ : vecteur de Γ.
$\Upsilon^{t_c \to t_r}$: champ de mouvement de t_c à t_r.
$\upsilon^{t_c \to t_r}$: vecteur de $\Upsilon^{t_c \to t_r}$.

$\psi_{j,\mathbf{m}}$: fonction d'ondelette dilatée du facteur 2^j et translatée au voisinage du point $2^j\mathbf{m}$.
$\phi_{j,\mathbf{m}}$: fonction d'échelle dilatée du facteur 2^j et translatée au voisinage du point $2^j\mathbf{m}$.
$d_j[\mathbf{m}]$: coefficient d'ondelette.
$a_j[\mathbf{m}]$: coefficient d'approximation.

$\mathbf{R}$: Débit.
$\mathbf{D}$: Distorsion.

Id : Identité.

$\hat{s}$: signal s décodé.

Maillage déformable

l_a : taille d'une arête.
N_s : nombre de sommets.
i : indice d'un sommet.
(x_i, y_i) : coordonnées du sommet i dans le domaine image.
(u_i, v_i) : coordonnées du sommet i dans le domaine texture.
$\mathcal{M}$: maillage dans le domaine image.

$\tilde{\mathcal{M}}$: maillage dans le domaine texture.
$\mathbf{E}_d$: énergie de déformation.
ω_d : poids associé à $\mathbf{E}_d$.
Q_m : pas de quantification pour le mouvement.
Q_g : pas de quantification pour la géométrie.

Analyse-Synthèse spatiales

I : image.
T : texture.
$\mathcal{D}$: domaine image.
$\tilde{\mathcal{D}}$: domaine texture.
(x, y) : point ou pixel dans le domaine image.
(u, v) : point ou pixel dans le domaine texture.
k : itération.
w : transformation spatiale.
w^{-1} : transformation inverse.
$\mathbf{C}$: coût de description de la texture.
$\tilde{T}_j$: approximation de T à l'échelle 2^j.
T_{cible} : texture cible.
J_w : jacobien de la déformation w.
$I^\star$: image de qualité maximale que l'on peut reconstruire sans perte sur la texture et la déformation w.
I_ϵ : image de résidu $I - I^\star$.
r_d : rapport entre les dimensions de la texture et les dimensions de l'image.
n_p : nombre de plans de bits non reconstruits pour la géométrie.
T_{ssim} : seuil utilisé pour détecter les zones texturées mal reconstruites.
T_w : seuil utilisé pour détecter les déformations de mailles non significatives.

Analyse-Synthèse spatio-temporelles

N_G : taille des GOF.
I_t : image à l'instant t.
$\mathcal{D}_t$: domaine image à l'instant t.
T_t : texture à l'instant t.
$\tilde{\mathcal{D}}_t$: domaine texture à l'instant t.
t_r : instant de référence pour une estimation de mouvement.
t_c : instant courant.
t_p : instant de projection.
$\bar{I}_t$: prédiction de I_t.
$\bar{I}_{t_c \to t_r}$: prédiction de I_{t_c} après compensation en mouvement de I_{t_r}.
I_{BF} : basse fréquence temporelle du GOF compensé en mouvement.
w^g_{BF} : géométrie calculée sur I_{BF}.

Introduction

Les images et les vidéos ont envahi notre quotidien. L'évolution des technologies numériques et le nombre toujours plus important de services proposés à l'utilisateur ont favorisé l'explosion des contenus multi-medias. Afin de stocker ces contenus, de les transférer ou de les diffuser en temps réel, une étape de **compression** est nécessaire : pour une capacité de stockage ou de débit en diffusion donnée, il faut fournir à l'utilisateur la meilleure qualité visuelle possible. Même si les capacités des disques durs et des réseaux se sont accrues, la problématique de compression reste plus que jamais pertinente. En effet, cet accroissement des capacités a aussi fait naître de nouvelles applications comme la TV sur mobile ou sur internet, la HD au format progressif 1080p, la TV3D...

Au-delà de la recherche du meilleur compromis débit-distorsion possible, on demande aujourd'hui aux algorithmes de compression une grande souplesse face à la nature variée des applications, des réseaux et des terminaux. On parle souvent de *convergence*. En particulier, la problématique de « scalabilité » s'est imposée comme un enjeu majeur au cours des dernières années : un flux encodé est dit « scalable » ou « emboîté » s'il peut être tronqué pour s'adapter à des capacités de débit ou des résolutions spatiales et temporelles d'affichage variées.

Pour la vidéo, le standard de compression scalable actuel est H.264/MPEG-4 SVC, amendement au standard non scalable H.264/MPEG-4 AVC. AVC est né d'un effort commun entre les deux organismes de standardisation que sont l'ITU-T et l'ISO/IEC. Il s'inscrit dans la lignée des standards H.26x et MPEG-x qui s'appuient sur un codage prédictif. Le principe est de prédire une image courante à l'aide d'une ou plusieurs images déjà encodées en effectuant une estimation puis une compensation en mouvement, puis de transmettre le résidu de prédiction. Chaque brique de transformée et d'encodage a été exploitée et optimisée au cours des deux décennies précédentes. Avec AVC les capacités ont encore été multipliées par deux.

Malgré les bonnes performances offertes par le schéma de codage prédictif, il est important de proposer des **approches en rupture** avec ce schéma et d'évaluer leur potentiel par rapport aux standards. Dans cette optique, les travaux de Cammas et Pateux [Cam04b, CP03b] ont abouti à un schéma de codage dit par **analyse-synthèse** dont le principe est illustré sur la figure 1. L'idée ici est de déformer le contenu d'un groupe d'images pour l'adapter à une décomposition le long de l'axe temporel fixe. Un suivi de mouvement par maillage déformable est appliqué puis, en s'appuyant sur le mouvement estimé, chaque image d'origine est « projetée » dans un même système de coordonnées. Le groupe d'images (GOF) compensé en mouvement est ensuite décomposé

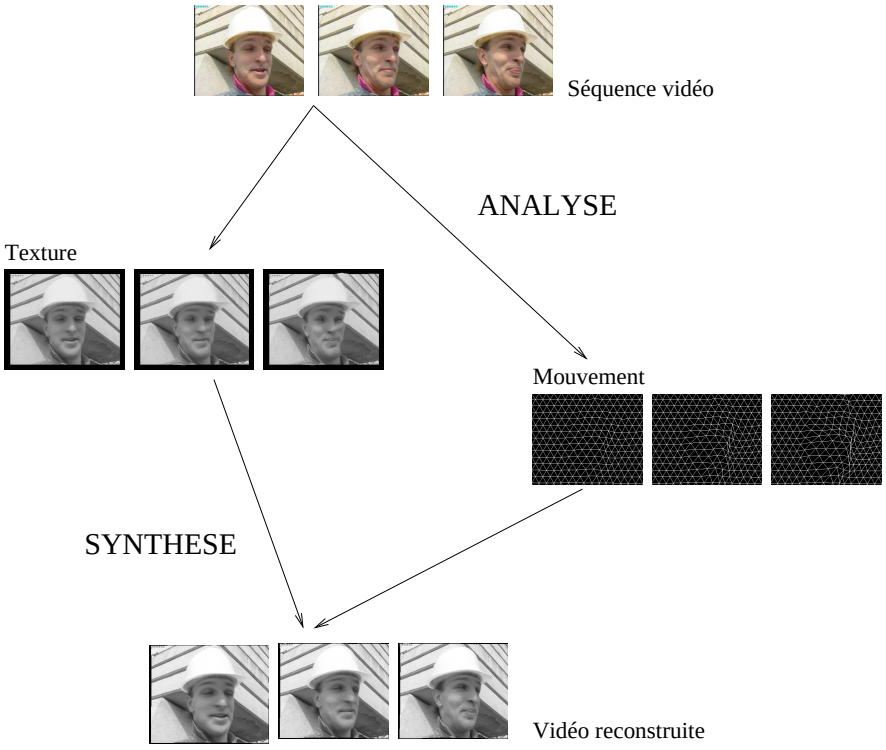


Fig. 1 : Schéma par analyse-synthèse temporelles proposé par Cammas et Pateux [Cam04b, CP03b].

par une ondelette 1D dans la direction temporelle et les sous-bandes temporelles générées sont envoyées à JPEG2000. Le mouvement doit être transmis pour synthétiser les images en bout de chaîne. Dans ce schéma, on remarque donc que **le contenu des images est adapté au noyau de décomposition temporelle**. C'est une distinction forte par rapport au schéma standard et aux schémas s'appuyant sur une transformée ondelette adaptée au mouvement (« Motion Compensated Temporel Filtering ») où c'est le noyau qui s'adapte au contenu temporel. Le schéma de Cammas et Pateux offre en outre une scalabilité naturelle qui a montré de bonnes performances par rapport au codeur SVC. Les travaux que nous avons menés dans cette thèse s'inscrivent dans la continuité de ce schéma.

Comme nous l'avons noté, dans le schéma précédent les sous-bandes temporelles sont codées par JPEG2000. JPEG2000 est le standard de compression scalable actuel pour l'image fixe. Ce standard est basé sur **la transformée en ondelettes** et le codeur EBCOT. JPEG2000 a fortement amélioré le compromis débit-distorsion par rapport au précédent standard JPEG à base de DCT, tout en offrant la scalabilité. Cependant, des améliorations sont possibles. En particulier, la transformée en ondelettes classiques opère un filtrage des images selon des directions fixes (l'horizontale et la verticale) souvent inadaptées au contenu local. Lorsque l'image contient des caractéristiques *géométriques* (contours, motifs de texture) non horizontales ni verticales, leur énergie se trouve répartie sur un nombre importants de coefficients dans le domaine ondelette. Lors d'une approximation non linéaire à l'aide d'un nombre limité de coefficients, ces coefficients ont une probabilité forte d'être seuillés, ce qui se traduit par des rebonds d'ondelettes gênants après reconstruction de l'image.

Pour remédier à ce phénomène, une seconde génération d'ondelettes est née. Le but est de proposer des bases ou des dictionnaires d'atomes de formes variées pouvant capturer les caractéristiques géométriques d'une image pour produire des représentations parcimonieuses. L'énergie d'un contour est alors concentrée sur un petit nombre de coefficients de forte énergie qui ne sont pas seuillés lors d'une approximation et permettent une meilleure reconstruction de la géométrie. Lorsque la base d'ondelettes est *adaptative*, les paramètres d'adaptation doivent être transmis avec les coefficients d'ondelettes pour pouvoir décoder l'image. La question est de savoir si le coût de codage de ces paramètres est compensé par la réduction de l'entropie des coefficients d'ondelettes.

Les premiers travaux que nous avons menés dans cette thèse concernent le codage d'images fixes. L'idée est d'exploiter une approche similaire à celle adoptée par Cammas et Pateux dans le cadre de la vidéo en proposant de **déformer le contenu spatial d'une image fixe pour l'adapter à un filtrage fixe horizontal-vertical**. Comme dans les travaux précédents, nous choisissons de modéliser la déformation par un maillage déformable. Le problème principal est de déterminer une heuristique qui permet de définir la position des noeuds du maillage, paramètres de déformation. Au chapitre 4, nous décrivons une technique d'estimation qui répond à ce problème. Elle s'appuie sur l'expression du coût de codage de l'image déformée en fonction des paramètres de déformation. A l'issue de cette analyse, l'image est représentée par une image déformée, appelée *texture*, de moindre coût de codage et par les paramètres de déformation. Après codage, transmission et décodage de ces informations, l'image d'ori-

gine peut être synthétisée en inversant la déformation. Ce schéma par analyse-synthèse spatiale est illustré sur la figure 2. Ses performances en termes de compression par rapport à JPEG2000 sont étudiées. Visuellement, on observe une meilleure reconstruction des contours des images avec une atténuation significative de l'effet rebond. Cependant, les métriques utilisées (PSNR et SSIM) donnent des résultats objectifs moins bons que ceux de JPEG2000. L'explication vient des pertes numériques introduites en ré-échantillonnant l'image lors de l'analyse puis de la synthèse. Ces pertes numériques sont surtout visibles dans les zones texturées. Des post-traitements à l'analyse sont alors proposés pour les limiter. Ils permettent de ré-hausser la qualité visuelle et objective des images reconstruites.

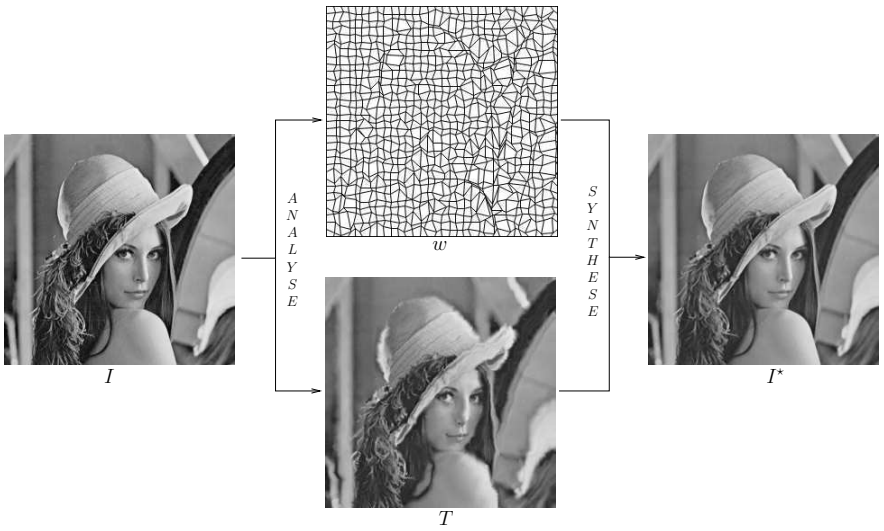


Fig. 2 : Schéma par analyse-synthèse spatiale proposé au chapitre 4.

Les seconds travaux que nous avons menés dans cette thèse portent sur le codage de vidéos. Ces travaux sont décrits au chapitre 5. L'idée est de fusionner les travaux de thèse de Cammas et nos travaux sur l'image fixe. En effet, dans l'approche de Cammas et Pateux, les images compensées en mouvement conservent des caractéristiques géométriques qui ne sont pas prises en compte. En adoptant l'analyse spatiale précédente, il est possible d'adapter les images d'origine d'un GOF à la fois à une décomposition temporelle fixe et à une décomposition spatiale fixe (horizontale-verticale). Cependant, estimer et transmettre des paramètres géométriques pour chaque image du GOF serait prohibitif. Nous proposons donc d'estimer une seule géométrie pour tout le GOF compensé en mouvement : la géométrie de la basse fréquence temporelle. Si l'alignement temporel a été efficace, alors toutes les images du GOF compensé ont une géométrie similaire à celle de l'image de basse fréquence temporelle. L'adaptation du contenu de chaque image d'origine peut ainsi se faire en appliquant une compensation en mouvement différente pour chaque image suivie d'une compensation en géométrie identique pour chaque image. Le groupe d'images déformées appelé *groupe de textures* est ainsi

adapté à une décomposition par ondelettes 3D. Comme pour l'image fixe, les paramètres de mouvement et de géométrie doivent être transmis avec les textures pour pouvoir reconstruire les images en bout de chaîne. La question est de savoir si la prise en compte de la géométrie apporte un gain par rapport au schéma d'analyse-synthèse temporelle. Bien qu'une seule géométrie soit transmise pour un GOF, nos résultats indiquent que cette géométrie occupe une part trop importante du débit si l'on souhaite extraire suffisamment de détails géométriques. Bien que la reconstruction des contours soit améliorée, la qualité visuelle générale des images est moins bonne. Notons cependant que le schéma proposé est un schéma général qui peut être appliqué avec d'autres modèles de mouvement et/ou de géométrie que le maillage déformable.

Le manuscrit est organisé comme suit :

Chapitre 1 Ce chapitre pose le cadre de notre travail. Il définit le contenu des images (mouvement et géométrie), s'intéresse à leur impact visuel puis se penche sur la problématique de représentation. Les limites des ondelettes séparables et l'intérêt des ondelettes seconde génération sont mis en avant. La dernière section est dédiée à la problématique de compression. Elle rappelle les briques de base d'un algorithme de compression d'images, énonce le problème de l'optimisation débit-distorsion puis fait un focus sur la scalabilité avant de décrire les principaux codeurs de sous-bandes d'ondelettes.

Chapitre 2 Ce chapitre propose un état de l'art sur les outils antérieurs permettant de prendre en compte le contenu spatial (géométrique) d'une image fixe. Dans un premier temps, les bases fixes sont traitées (Ridgelets, Curvelets, Contourlets...). Les deux sections suivantes sont consacrées aux méthodes adaptatives qui s'appuient sur un modèle de géométrie à transmettre. Nous nous penchons tout d'abord sur les outils permettant une analyse locale (lattices, lifting directionnel, déformation de blocs, Bandelettes...). Puis nous étudions les modèles de représentation globaux. Le maillage et ses propriétés sont en particulier introduits.

Chapitre 3 Ce chapitre fait écho au chapitre précédent en décrivant les outils antérieurs permettant une adaptation au contenu temporel dans une vidéo. La première section décrit différents modèles de mouvement (« Block Matching », maillage déformable...). La seconde section s'intéresse à la manière d'estimer les paramètres de ces modèles. Enfin, la troisième partie décrit différentes façon d'exploiter le mouvement dans un algorithme de compression. Nous revenons ainsi par exemple sur le codage prédictif et sur le schéma par analyse-synthèse temporelles de Cammas et Pateux.

Chapitre 4 Ce chapitre présente le travail que nous avons mené sur l'image fixe. La première section décrit le principe général du schéma par analyse-synthèse spatiales noté AS2D. Nous distinguons notre travail de l'art antérieur et introduisons le maillage déformable comme modèle de géométrie. La seconde section se consacre à l'analyse. En

particulier, nous définissons le coût de description de la texture à minimiser, nous l'exprimons par rapport aux paramètres de déformation puis proposons une technique d'optimisation ressemblant fortement à une estimation de mouvement entre deux images. La section 1.4 décrit la façon dont nous codons la texture et le maillage et présente des premiers résultats de compression obtenus en utilisant des mailles avec une taille de l'ordre de 16×16 pour modéliser la géométrie. Cette taille de maille permet d'améliorer la qualité visuelle des images possédant une géométrie simple mais est insuffisante pour des images au contenu géométrique plus fin. Dans la section section 4.4, nous proposons d'apporter quelques modifications au schéma pour d'une part améliorer la qualité des zones texturées par rapport au schéma de base et d'autre part modéliser des contenus géométriques plus complexes.

Chapitre 5 Ce chapitre présente le travail que nous avons mené sur la vidéo. La première section décrit le schéma général d'analyse-synthèse spatio-temporelles noté AS2D+t. La seconde section montre les résultats obtenus en utilisant le maillage déformable à la fois comme modèle de géométrie et comme modèle de mouvement. Des résultats comparatifs avec le standard H.264/MPEG-4 SVC et le schéma d'analyse-synthèse temporelle AS t sont donnés. Les résultats donnés par SVC sont meilleurs que ceux donnés par nos implémentations des schémas par analyse-synthèse. D'autre part, les résultats indiquent que le coût de la géométrie dans le schéma AS2D+t est trop important pour améliorer les performances du schéma AS t. Dans la dernière section, nous avons cherché à améliorer l'alignement temporel des images en utilisant des modèles de mouvement moins contraints que le maillage déformable permettant de représenter des discontinuités de mouvement. Ces modèles permettent effectivement un meilleur alignement temporel mais les résultats de codage obtenus n'apportent pas d'amélioration significative par rapport à ceux obtenus avec un maillage déformable. De surcroît, autoriser les discontinuités de mouvement engendre des zones non connectées à la synthèse dont la reconstruction est un problème ouvert.

Suite au chapitre 5, nous donnons les conclusions de nos travaux en rappelant les principales contributions. Dans les perspectives, nous introduisons une nouvelle structure pour représenter une vidéo. Elle sera étudiée dans des travaux de thèse futurs.

Chapitre 1

Cadre de travail

Exploiter les redondances d'un signal est un principe de base en compression. Lorsque le signal est une image ou une vidéo, certaines connaissances a priori peuvent guider la recherche des redondances. En particulier, la *géométrie* en 2D et le *mouvement* le long de l'axe temporel définissent des trajectoires régulières dont on peut tirer avantage. S'intéresser à ces informations de *structure* est d'autant plus important qu'elles jouent un rôle primordial dans l'interprétation des images par le Système Visuel Humain (SVH).

Dans ce chapitre, nous introduisons ces notions qui sont étroitement liées au sujet de thèse et faciliteront la lecture des chapitres suivants. Après une courte description du contenu des images, nous nous arrêtons en section 1.2 sur le processus de construction mental des images par le SVH et sur la notion cruciale de *qualité*. La section 1.3 est quant à elle dédiée à la problématique de *représentation*. Un focus sur les ondelettes nous permet de préciser les motivations qui ont conduit à l'étude sur les ondelettes seconde génération. Enfin, en section 1.4, nous rappelons certains enjeux spécifiques à la *compression* et évaluons les performances des codeurs ondelettes non adaptatifs face à ces enjeux. Notons que différents travaux de thèse ont abordé un ou plusieurs de ces thèmes précédemment, par exemple [Pen02, Cha05b, Pey05b, Vel05b]. Des descriptions mathématiques plus approfondies pourront être trouvées dans ces ouvrages.

1.1 Contenu des images

1.1.1 Contenu spatial et flux géométrique

Une image naturelle est une projection 2D d'une scène à un instant donné. Son intensité peut être modélisée par une fonction bidimensionnelle continue I définie sur un intervalle borné $\mathcal{D}$. La valeur de cette fonction en un point $\mathbf{x} = (x, y)$ dépend principalement de la quantité de lumière réfléchie par les objets de la scène, mais également des bruits d'acquisition. Cette thèse s'intéresse en particulier aux images discrètes définies sur une grille de pixels. Les dimensions de cette grille déterminent la *résolution* de l'image. Au cours de ce manuscrit, nous serons parfois amenés à considérer une image I comme une *surface* donnée par l'ensemble des points 3D $\{(x, y, I(x, y))\}_{(x,y)\in\mathcal{D}}$.

Les images naturelles que l'on croise dans notre quotidien ne sont pas des bruits purement aléatoires. Elles véhiculent une information qui est portée essentiellement par trois éléments :
Les contours. Un contour est formé lorsque deux objets de la scène se superposent ou lorsque deux zones contiguës d'un même objet ont des niveaux de gris très différents. A l'échelle du pixel, le passage d'un objet à un autre dans une direction donnée se caractérise par une modification abrupte du niveau de gris nommée discontinuité de type point. A l'échelle de l'image, les discontinuités de type point se regroupent pour former une discontinuité, ou *singularité*, 1D que l'œil reconnaît comme un contour.
Les zones texturées. Les zones texturées sont des zones de l'image comportant des motifs fins qui se reproduisent à l'échelle du pixel selon un schéma déterministe ou stochastique [EF01, HB95]. La définition d'une zone texturée est dépendante de la résolution de l'image : un motif dans une zone texturée peut apparaître comme un objet à part entière délimité par des contours à une résolution plus importante.
Les zones homogènes. Ces zones sont des régions de l'image où le niveau de gris varie de façon régulière. Une image de type « cartoon » est composée principalement de zones homogènes délimitées par des contours.
Comme nous pouvons l'observer, ces trois éléments porteurs d'information contiennent chacun une dose de régularité plus ou moins complexe à représenter.

Nous définissons le *flux géométrique* Γ comme l'ensemble des vecteurs $\gamma(\mathbf{x})$ donnant la direction de régularité maximale en chaque point $\mathbf{x}$ du domaine image. Une *ligne de flux géométrique* s'obtient en intégrant le flux de proche en proche lorsque cela est possible [MAD05]. La taille d'une ligne de flux dépend de la régularité des variations du flux. Les lignes seront donc plus grandes dans les zones homogènes et contenant un contour que dans les zones texturées où le flux est plus chaotique. Si la géométrie est définie comme l'ensemble de ces lignes de flux, elle possède donc un caractère *multi-échelles* : selon le contenu d'une image, elle peut être capturée à l'échelle du pixel, de quelques pixels ou à l'échelle de l'image lorsqu'un objet occupe la surface du domaine. Notons que dans une zone homogène caractérisée par une régularité isotrope (à savoir une zone régulière dans toutes les directions), le flux n'est pas défini de manière unique.

1.1.2 Contenu temporel et flux optique

Une vidéo est une projection 2D d'une scène qui évolue dans le temps. Les variations d'intensité dans le temps sont dues au mouvement *réel* des objets dans l'espace, mais également au mouvement de la caméra et aux changements d'illumination : on parle de *mouvement apparent*. Une vidéo discrète est une séquence d'images acquises à instants réguliers. Dans la suite, une image particulière de cette séquence à l'instant t discret sera notée I_t et son domaine de définition $\mathcal{D}_t$.

En se basant sur les variations de l'intensité entre un instant t et un instant t', on peut définir en chaque point $\mathbf{x}$ de $\mathcal{D}_t$ un vecteur mouvement $v(\mathbf{x})$ associant $\mathbf{x}$ à un point dans $\mathcal{D}_{t'}$. L'ensemble de ces vecteurs sera appelé *flux optique* ou *champ de mouvement* et sera noté Υ. Chaque vecteur $v(\mathbf{x})$ donne la direction de régularité temporelle de la vidéo au point $\mathbf{x}$ entre t et t'. Une *ligne de flux temporelle* s'obtient en intégrant le flux

optique de proche en proche lorsque cela est possible (figure 1.1). La taille d'une ligne de flux dépend de la régularité des variations du flux dans le temps. Le mouvement possède donc également un caractère multi-échelles. L'intégration d'une ligne de flux s'arrête lorsque le flux optique est discontinu. Ceci arrive en particulier lorsqu'une zone de l'image apparaît ou disparaît entre l'instant t et l'instant t'. On parle de *zone à occultation*.

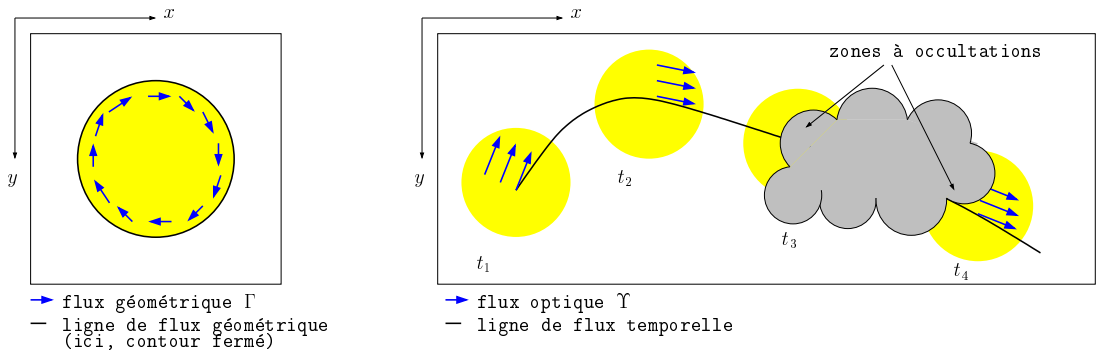


Fig. 1.1 : Flux géométrique, flux optique et lignes de flux.

1.2 Enseignements de la vision

La problématique de compression est avant tout une problématique de représentation. Il s'agit de déterminer l'approximation d'un signal ayant la meilleure *qualité* possible pour un nombre fixé de coefficients. Comme les images et les vidéos sont des signaux destinés à être visualisés, l'œil humain est le seul juge pertinent de cette qualité. L'étude de la perception visuelle est donc essentielle pour savoir comment l'homme se construit une représentation mentale du monde qui l'entoure et ainsi identifier les caractéristiques les plus importantes dans une image. Elle offre des pistes pour élaborer de nouvelles représentations et de nouvelles méthodes d'évaluation.

1.2.1 Représentation des scènes naturelles

L'étude de la vision humaine peut être abordée de différentes façons. Il y a tout d'abord les approches basées sur des spéculations théoriques, comme le modèle fondateur proposé par David Marr [Mar82] au début des années 80. Selon ce modèle, dit *constructiviste*, la reconnaissance d'un objet 3D par le SVH suit un processus itératif « bottom-up » en 3 temps illustré sur la figure 1.2. Les contours jouent un rôle primordial car ils sont détectés en premier (stade « primal »). Les surfaces et orientations ne sont détectées que dans un second temps (stade « 2D+1/2 »). Chaque étape génère des signaux qui sont mis en correspondance avec des modèles ou « patterns » présents en mémoire (d'après ce modèle, la comparaison pixel à pixel serait un processus bien trop complexe, même pour le cerveau humain !). Les suggestions de la mémoire sont

discriminées en suivant une approche « top down ». Ces travaux suggèrent l'importance des contours dans le processus de représentation mentale des objets. La théorie plus ancienne de la Gestalt [Wer38] avait déjà mis en évidence l'importance des contours dans la perception. En particulier, le principe de « bonne continuation » semble suggérer que le SVH opère un processus d'intégration du flux géométrique qui lui permet de capturer les régularités le long des contours, et parfois même au-delà (illusions d'optique).

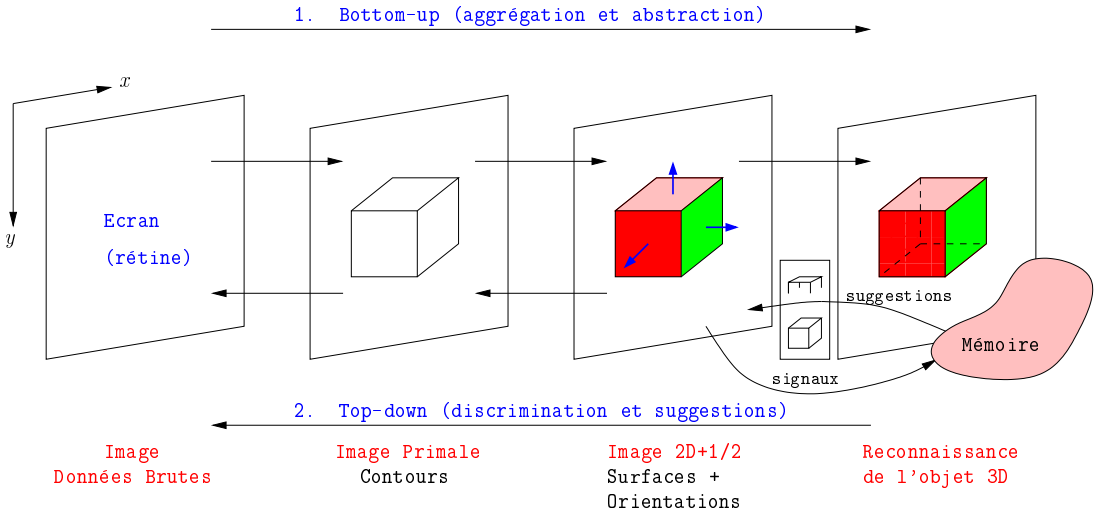


Fig. 1.2 : Approche fondatrice de David Marr [Mar82], dite *constructiviste*.

Il y a ensuite les approches pratiques qui se basent sur des systèmes sophistiqués d'imagerie du cerveau humain ou sur l'implantation d'électrodes sur des animaux. Le but est d'analyser le comportement des différentes aires du cortex visuel face à des stimuli pour comprendre quels types de signaux élémentaires sont utilisés par le SVH pour construire une représentation globale des objets. En particulier, les travaux de Field et al. [Fie87, Fie93, FHH93] ont montré que la réponse des neurones (en particulier dans la région V1 du cortex) était très sensible à la position et à l'échelle d'un stimulus. Les observations de l'auteur semblent suggérer d'une part que la vision est un phénomène naturellement *multi-échelles* et d'autre part que la réponse des neurones de la région V1 à un stimulus a des propriétés très comparables à celles d'une *ondelette* (voir paragraphe 1.3.5). D'autres études empiriques [OF96, vHvdS98] ont ensuite montré que la réponse des neurones est aussi très sensible à l'orientation et l'élongation du stimulus. Ceci suggère que les éléments de base permettant une représentation compacte d'une scène naturelle sont fortement directionnels, contrairement aux ondelettes. Enfin, des expériences d'imagerie du cerveau [MDF+99, Wan95], basées sur l'observation par IRM fonctionnelle de la réponse neuronale à des images contenant des courbes allongées, ont mis en avant une forte activité dans la région V3 du cortex, comme si une tâche d'intégration complexe s'y déroulait. Ce résultat empirique est donc à mettre en relation avec le modèle théorique proposé par la Gestalt.

D'après les études sur la vision, il semble évident que la représentation pixel à pixel d'une image ne correspond pas au processus de construction exercé par l'œil humain. L'œil humain ne « voit » pas l'image pixel à pixel : il capture les régularités à différentes échelles. Si l'on veut s'inspirer de la vision pour bâtir une représentation mathématique d'une image, on voit donc qu'il faut définir des fonctions élémentaires de type ondelettes possédant des positions, échelles et orientations variées.

1.2.2 La notion de qualité

La qualité d'une image ou d'une vidéo est une notion hautement subjective. Le résultat de la perception est propre à chacun et dépend de nombreux facteurs tel le niveau d'attention, l'état émotionnel ou le vécu de la personne. Les éléments stockés en mémoire influencent l'interprétation en complétant la perception par des images et des souvenirs. Dans un cadre de compression avec pertes, les images sont susceptibles d'être dégradées et il est donc nécessaire d'évaluer leur qualité. Cette évaluation peut être faite à l'aide de tests subjectifs suivant un protocole bien défini [BT.02] pour aboutir à une note MOS (« Mean Opinion Score »). Cependant, ces tests sont très coûteux en temps et il est donc plus pratique d'évaluer la qualité avec des métriques objectives.

Les métriques objectives peuvent être groupées en trois grandes classes, selon que l'image originale (non dégradée) servant de référence pour la comparaison est disponible ou non. La plupart des approches existantes sont dites à *référence complète* et supposent que l'image de référence est connue. En pratique, lorsqu'un contrôle de qualité est nécessaire dans une chaîne de transmission, l'image de référence n'est en général pas disponible. Certaines de ses caractéristiques peuvent être extraites au moment de l'encodage puis transmises afin de permettre une évaluation dite à *référence réduite* (RR) [CCB03, CVGPC06]. Dans le cas extrême où aucune information n'est disponible, une évaluation dite *sans référence* ou « blind » (NR) est requise [YWCW05, FK05]. En général les métriques RR et NR se concentrent sur des artefacts particuliers comme les phénomènes de blocs ou de rebonds. Dans notre cadre expérimental, nous supposerons que les images d'origine sont disponibles lors de l'évaluation des images dégradées et nous utiliserons donc une métrique à référence complète.

La métrique à référence complète la plus simple et la plus largement utilisée est le PSNR (« Peak Signal to Noise Ratio »). Elle se base sur l'erreur quadratique moyenne (EQM), calculée en moyennant l'énergie du résidu entre l'image d'origine I et l'image dégradée $\tilde{I}$:

$$PSNR(\tilde{I}, I) = 10 \log_{10} \left[\frac{MAX^2}{EQM(\tilde{I}, I)} \right] \tag{1.1}$$

où $MAX = 255$ si les valeurs de l'image sont codées sur 8 bits. Le PSNR a plusieurs avantages : il est simple à calculer, possède une signification physique claire et comme nous le verrons l'EQM est très pratique dans un contexte d'optimisation mathématique. Cependant, le PSNR ne correspond pas bien à la qualité visuelle perçue [Gir93, WBL02]. En effet, l'EQM compare les images pixel à pixel. Or, nous avons vu plus haut que l'œil humain est sensible aux structures géométriques et temporelles des images. En outre, l'œil humain agit comme un filtre lissant sur les données brutes d'une image. C'est ce

qu'on appelle le phénomène de *masquage*. Pour s'en convaincre, il suffit d'observer la surface d'une image naturelle dans un espace 3D (voir figure 1.3). On s'aperçoit que les données brutes sont bien plus bruitées qu'il n'y paraît en regardant l'image. Cela est dû à ce phénomène de masquage. C'est ce même phénomène qui empêche l'œil humain de distinguer des différences de niveau de gris inférieures à un certain seuil.

(a) (b) (c)

Fig. 1.3 : Effet de masquage opéré par l'œil humain. (a) Image *Lena I* d'origine, (b) Points $\{(x, y, I(x, y))\}_{(x,y)\in\mathcal{D}}$ dans l'espace 3D, point de vue de l'image, (b) Vue de côté de la surface. L'œil humain ne discerne pas les pics (très hautes fréquences ou bruits d'acquisition).

Depuis plusieurs années, de gros efforts ont été consentis pour développer des métriques tenant compte des caractéristiques du SVH (un état de l'art de ces métriques pour l'image fixe et la vidéo est présenté dans [PS00a, EB98, WSB03]). La majorité de ces méthodes proposent de modifier l'EQM pour pénaliser les erreurs selon leur visibilité. Récemment Wang et al. ont introduit une nouvelle métrique SSIM (« Structural Similarity Image Metric ») pour l'évaluation d'images fixes [WBSS04] et de videos [WLB04]. Elle intègre l'hypothèse que le SVH extrait les caractéristiques structurelles d'une image à partir du flux géométrique et montre une bonne corrélation avec le MOS comparé à d'autres mesures. Nous l'utiliserons donc dans certains de nos résultats pour pondérer le PSNR. Notons cependant que l'élaboration d'une métrique adaptée à la perception reste un problème ouvert et d'autant plus important que l'évaluation des algorithmes en dépend.

1.3 Représentation

La représentation d'une image numérique par ses données brutes (niveaux de gris) n'est pas pertinente pour le codage car elle ne prend pas en compte la corrélation entre un pixel et son voisinage. Or, réduire la corrélation est essentiel dans un cadre de compression ou d'approximation. Dans cette section, nous nous penchons donc sur cette problématique de représentation.

1.3.1 Analyse-Synthèse

Considérons l'ensemble des fonctions discrètes de carré intégrable et définies sur un domaine $\mathcal{D} \in \mathbb{Z}^d$ où d est une dimension fixée. Cet ensemble est noté $\mathcal{L}_2(\mathcal{D})$. C'est un

espace vectoriel muni du produit scalaire $<,>$ défini par :

$$< f, g >= \sum_{\mathbf{x}\in\mathcal{D}} f(\mathbf{x})g^*(\mathbf{x}) \quad \forall (f,g) \in \mathcal{L}_2(\mathcal{D}) \tag{1.2}$$

La notation * désigne le complexe conjugué. Dans la suite nous restreignons $\mathcal{L}_2(\mathcal{D})$ à l'ensemble des fonctions à valeurs dans $\mathbb{R}$. Avec ce produit scalaire, l'énergie d'une fonction $f \in \mathcal{L}_2(\mathcal{D})$ s'écrit :

$$\|f\|^2 = \sum_{\mathbf{x}\in\mathbb{Z}^2} < f, f > \tag{1.3}$$

La représentation d'une fonction fait appel à des briques élémentaires qui permettent l'*analyse* et la *synthèse* du signal. Soit $\mathcal{F} = \{\psi_m\}_m$ une famille de fonctions élémentaires génératrice de $\mathcal{L}^2(\mathbb{Z}^2)$. L'*analyse* d'une fonction f par $\mathcal{F}$ est réalisée en calculant les produits scalaires de f avec chaque brique élémentaire ψ_m. Ces projections donnent une suite de coefficients $\{c_m =< f, \psi_m >\}_m$. La question est de savoir si la seule donnée de ces coefficients permet de caractériser f et de la reconstruire. C'est le cas si $\mathcal{F}$ est une *frame*, c'est-à-dire si et seulement si il existe deux constantes K_1 et K_2 strictement positives telles que, pour toute fonction $f \in \mathcal{L}_2(\mathcal{D})$:

$$K_1\|f\|^2 \leqslant \sum_m | < f, \psi_m > |^2 \leqslant K_2\|f\|^2 \tag{1.4}$$

Un tel encadrement montre que la suite des produits scalaires caractérise f de façon stable. Il signifie aussi que l'opérateur d'analyse qui associe à f la suite $\{c_m\}_m$ est inversible à gauche. On peut donc construire une deuxième famille $\tilde{\mathcal{F}} = \left\{\tilde{\psi}_m\right\}_m$ appelée frame duale qui permet la *synthèse* de f par la formule de reconstruction :

$$f = \sum_m c_m\tilde{\psi}_m \ = \sum_m < f, \psi_m > \tilde{\psi}_m \tag{1.5}$$

En ce sens, la suite de coefficients $\{c_m\}_m$ est bien une *représentation* de f car sa connaissance est formellement équivalente à celle de f. Notons que la formule (1.5) ne correspond pas forcément à la décomposition de f dans une *base* de fonctions. En effet, dans le cas général, une frame est une famille liée qui aboutit donc à une représentation redondante : le nombre de briques élémentaires ψ_m nécessaire et suffisant pour représenter toute fonction $f \in \mathcal{L}_2(\mathcal{D})$ est supérieur au nombre d'échantillons dans $\mathcal{D}$. Dans ce cas, le *facteur de redondance* r est simplement le rapport entre les deux nombres. Par comparaison, une *base* est une frame qui en plus est une famille libre. La représentation dans une base est dite à échantillonnage critique car le nombre de briques élémentaires ψ_m nécessaire et suffisant pour représenter toute fonction $f \in \mathcal{L}_2(\mathcal{D})$ est égal au nombre d'échantillons dans $\mathcal{D}$. Si les fonctions de base sont orthogonales alors $K_1 = K_2$ dans l'expression (1.4). Si elles sont orthonormales, on obtient l'égalité de Parseval :

$$\sum_m c_m^2 = \sum_{\mathbf{x}\in\mathcal{D}} f^2(\mathbf{x}) \tag{1.6}$$

1.3.2 Approximation non linéaire

Soit $\mathcal{B} = \{\psi_m\}_m$ une base orthonormée de $\mathcal{L}_2(\mathcal{D})$. La somme partielle

$$\tilde{f}_M = \sum_{m \in I_M} < f, \psi_m > \tilde{\psi}_m, \tag{1.7}$$

est une approximation de f obtenue en ne retenant que M projections. I_M donne les indices des coefficients retenus. L'orthonormalité de la base permet d'exprimer facilement l'erreur quadratique d'approximation :

$$\|f - \tilde{f}_M\|^2 = \sum_{m \notin IM} | < f, \psi_m > |^2 \tag{1.8}$$

Une approximation *linéaire* se calcule en fixant arbitrairement le jeu d'indices I_M. Une approximation *non linéaire* [DeV98] se calcule en déterminant I_M de manière adaptative pour minimiser l'erreur d'approximation. Dans le cas d'une base orthonormale, le choix se simplifie grandement car la meilleure approximation non linéaire est obtenue en retenant les M coefficients de plus grande amplitude. Cette simplicité fait de la base orthonormale un outil de représentation privilégié.

Dans [PM05], Le Pennec et Mallat formulent le problème de représentation de la façon suivante. Pour une classe de fonctions particulière, il s'agit de déterminer la base orthonormée de représentation qui fournit la meilleure décroissance de l'erreur d'approximation non linéaire avec M coefficients lorsque M augmente. C'est le cas s'il existe une constante K et un coefficient α tels que :

$$\|f - \tilde{f}_M\|^2 \leqslant K \cdot M^{-\alpha} \tag{1.9}$$

où K est une constante qui ne dépend que de f. Pour avoir un taux de décroissance α élevé, il faut donc que l'énergie du signal soit concentrée sur un petit nombre de coefficients. Dans ce cas, la représentation est dite *compacte*, *creuse* ou bien encore *parcimonieuse*. Cette approche du problème de représentation est très intéressante car elle permet d'établir la borne théorique d'une représentation pour une certaine classe de signaux et donc de comparer deux représentations.

Dans le cas de la représentation d'images, les travaux théoriques se concentrent souvent sur les images composées de zones homogènes de régularité $\mathcal{C}^\alpha$ (c'est à dire α fois continues et dérivables) séparées par des discontinuités 1D de régularité $\mathcal{C}^\alpha$. Nous désignerons l'ensemble de ces images par $\mathcal{C}^\alpha \backslash \mathcal{C}^\alpha$. Si une image appartient à cet ensemble, alors la régularité α détermine le taux de décroissance optimal [PM05]. La recherche d'une meilleure représentation a donc l'objectif d'atteindre ce taux optimal. Avant de présenter les résultats obtenus par différentes représentations dans ce chapitre et le suivant, notons que les bornes théoriques d'approximation sont établies pour des classes d'images bien particulières $\mathcal{C}^\alpha \backslash \mathcal{C}^\alpha$ pour un α donné. Si les propriétés de l'image naturelle à approximer s'écartent de cette modélisation, il y a de fortes chances pour que la représentation comporte des résidus de corrélation. Dans un cadre de compression, nous verrons que la forme de ces résidus joue un rôle important. Notons enfin que dans

le cas d'une frame redondante, l'approximation non linéaire a aussi un sens du fait de la propriété de conservation d'énergie (1.4). Même si la recherche de $\tilde{f}_M$ dans ce cas est plus complexe, il n'est pas exclu qu'elle aboutisse à de meilleurs résultats d'approximation qu'une base orthonormée lorsque $M < N$.

1.3.3 Représentation en fréquence : Fourier

Au XIX^e siècle, Joseph Fourier découvre que tout signal périodique peut être représenté par une somme pondérée de sinusoïdes dont les poids constituent une série de Fourier. Ce résultat pose les bases de l'analyse harmonique. La transformée de Fourier permet de le généraliser à toutes fonctions intégrables. En dimension 1, la transformée de Fourier d'une fonction intégrable $f \in \mathcal{L}(\mathbb{R})$ s'écrit $\mathfrak{f}(\omega)$:

$$\mathfrak{f}(\omega) = \int_x f(x) \mathrm{e}^{-i\omega x} \mathrm{d}x \tag{1.10}$$

où ω détermine la pulsation de l'harmonique sur laquelle est projeté le signal. En dimension d la formule est la même mais x est remplacé par un vecteur $\mathbf{x}$ de dimension d. En dimension 2, les fonctions de base $\mathrm{e}^{-i(\omega_1 x + \omega_2 y)}$ peuvent s'écrire en coordonnées polaires :

$$\mathrm{e}^{-i(\omega_x x + \omega_y y)} = \mathrm{e}^{i\rho(x\cos\theta + y\sin\theta)} \tag{1.11}$$

avec $\rho = \sqrt{\omega_x^2 + \omega_y^2}$. Grâce à cette écriture, on voit donc que les briques de base servant à l'analyse d'une image sont les ondes planes qui se propagent dans la direction de θ en oscillant à la fréquence ρ (voir figure 1.4).

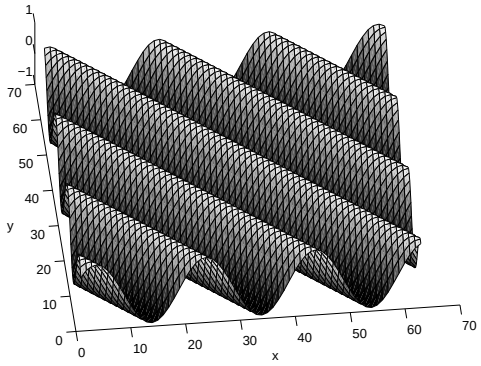


Fig. 1.4 : Partie réelle d'un noyau de Fourier. L'onde plane se propage ici dans la direction $\theta = 45^\circ$ ($\omega_x = \omega_y$).

Dans la pratique, la transformée de Fourier discrète d'une image de dimension $N \times N$ s'obtient en étendant le signal par périodisation le long des lignes et des colonnes puis en la projetant sur la famille de fonctions :

$$\left\{\psi_{k_x,k_y}(x,y) = \mathrm{e}^{\frac{i2\pi}{N}(k_x x + k_y y)}\right\}_{0 \leqslant k_x, k_y < N} \tag{1.12}$$

qui constitue une base orthogonale de l'espace des images périodiques de période N le long de leurs lignes et de leurs colonnes. Cette famille de N^2 vecteurs discrets est le produit séparable de deux bases de Fourier monodimensionnelles discrètes $\left\{\mathrm{e}^{i2\pi kx/N}\right\}_{0 \leqslant k < N}$.

Dans le cas général où $f(0) \neq f(N-1)$ le long d'une ligne ou d'une colonne, la périodisation de f crée des discontinuités spatiales abruptes qui se traduisent par un plus grand nombre de coefficients non nuls lors du passage dans le domaine de Fourier. Pour y remédier, chaque ligne et colonne de f peut être symétrisée de sorte que la périodisation du nouveau signal ne génère plus de discontinuité. Ce principe est à l'origine de la transformée en cosinus discrets (DCT).

Limite de la représentation fréquentielle. Conformément à la formule 1.10, un coefficient de Fourier est calculé en utilisant les valeurs de f sur l'ensemble de son support. L'observation des coefficients de Fourier permet donc de décrire un signal en termes de régularité *globale*. Dans le cas des signaux non stationnaires, cette représentation n'est pas économique. Un signal constant partout sauf en une discontinuité localisée est par exemple représenté par un grand nombre d'harmoniques. Ceci conduit à considérer à tort le signal comme un signal globalement peu régulier. En outre, puisque l'énergie de la discontinuité se trouve propagée sur un grand nombre d'harmoniques, approximer le signal en tronquant certaines fréquences conduit à un artefact visuel connu sous le nom de phénomène de Gibbs : des oscillations correspondant aux harmoniques tronquées apparaissent autour de la discontinuité (voir zoom figure 1.5).

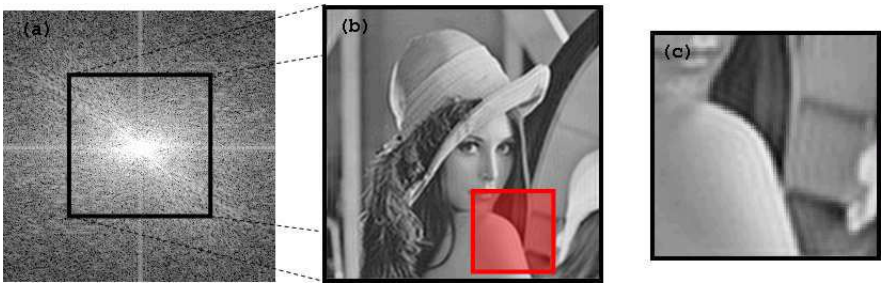


Fig. 1.5 : Spectre de Fourier et phénomène de Gibbs. (a) Spectre (amplitude) de *Lena*, (b) Approximation en tronquant les hautes fréquences, (c) Phénomène de Gibbs observé près des contours.

Cette limite de la représentation se retrouve dans le taux de décroissance de l'EQM lors d'une approximation non linéaire. En effet, pour une image de type $\mathcal{C}^\alpha \backslash \mathcal{C}^\alpha$, la DCT donne une erreur du type [Can98, CD99a] :

$$\|f - \tilde{f}_M\|^2 \leqslant K \cdot M^{-1/2} \tag{1.13}$$

quelque soit la régularité α.

Pour mieux représenter des signaux quelconques, il faut donc être capable de décrire des caractéristiques spatio-temporelles globales mais aussi locales. Le souhait d'une meilleure *localisation* a motivé la construction des représentations temps-fréquence.

1.3.4 Représentation temps-fréquence

Une idée simple pour créer des atomes à la fois localisés en temps et en fréquence est de multiplier une harmonique $e^{i\omega x}$ par une fonction fenêtre $g(x)$ bien localisée spatialement, par exemple une gaussienne comme le propose D. Gabor dans les années 1950, et ses versions translatées $g(x-m)$, $m \in \mathbb{R}$. La projection de f sur ces nouveaux atomes aboutit à la transformée de Fourier à fenêtre glissante :

$$\mathfrak{f}(\omega, m) = \int_{-\infty}^{+\infty} f(x) g(x-m) \mathrm{e}^{-i\omega x} \mathrm{d}x \tag{1.14}$$

D'après la formule (1.14), $\mathfrak{f}$ peut être vue de manière équivalente comme la transformée de Fourier de la fonction $f(x)g(x-m)$. Ce point de vue est d'ailleurs privilégié dans la pratique car il est plus simple de fenêtrer le signal et de lui appliquer une transformée de Fourier rapide. Notons cependant que, selon le principe d'incertitude d'Heisenberg, il n'est pas possible d'obtenir une localisation à la fois temporelle et fréquentielle arbitrairement précise. En effet, le support du spectre fréquentiel de l'atome temps-fréquence est d'autant plus large que son support temporel est compact. Ainsi, une harmonique permet une localisation fréquentielle infinie (un seul pic de fréquence dans le domaine de Fourier) mais une localisation temporelle nulle. Pour un atome de Gabor, la taille du support temporel est inversement proportionnelle à la taille du support fréquentiel. Le choix de la fenêtre $g(x)$ détermine donc le compromis entre localisation temporelle et fréquentielle. Souvent, les atomes de représentation sont schématisés comme des rectangles dans un plan temps fréquence repéré par les axes (ω, x) [Mal99]. Ces rectangles sont nommés boîtes de Heisenberg car leur aire minimale est imposée par le principe d'incertitude.

Comme précédemment, dans le cas 2D la transformée de Fourier à fenêtre glissante est une transformée séparable obtenue en réalisant des transformées 1D successives le long des lignes et des colonnes. Une telle transformée est présente dans le standard de compression d'images JPEG [Wal91] par exemple. Elle consiste à découper une image en blocs de taille 8×8 puis à effectuer une DCT sur chaque bloc.

Limite de la représentation temps-fréquence. La transformée à fenêtre glissante n'apporte qu'une réponse limitée au problème de double localisation car la forme de l'atome de base est fixe et arbitraire. Or, les signaux naturels comportent souvent des composantes de natures diverses : composantes régulières (basses fréquences) nécessitant une analyse plus globale et composantes moins régulières nécessitant une analyse plus locale. Le pavage régulier du plan temps-fréquence n'est donc pas optimal pour représenter de tels signaux et ne permet pas d'améliorer le taux de décroissance d'EQM lors d'une approximation non linéaire. En outre, le découpage d'une image en blocs produit un nouvel artefact lors d'une approximation connu sous le nom de phénomène

de blocs.

Pour trouver une meilleure représentation, on voit qu'il est nécessaire de construire des noyaux capables de capturer les caractéristiques multi-échelles d'un signal. Ceci nous conduit aux ondelettes.

1.3.5 Les Ondelettes

1.3.5.1 Bases 1D

Une famille d'ondelettes est obtenue au moyen de dilatations et de translations d'une fonction ψ élémentaire, appelée *ondelette mère*. Un noyau d'ondelette s'écrit de façon générale :

$$\psi_{a,b}(t) = a^{-1/2}\psi(\frac{t-b}{a}), \tag{1.15}$$

où $a > 0$ est le facteur de dilatation ou facteur d'échelle de l'ondelette et $b \in \mathbb{R}$ le facteur de translation. L'ondelette mère ψ possède deux caractéristiques importantes [Pey05b] :
La régularité d'ordre p. ψ possède un nombre $p \geqslant 1$ de moments nuls, c'est-à-dire que l'on a :

$$\int_t \psi(t)t^k \mathrm{d}t = 0 \quad \forall k \leqslant p-1 \tag{1.16}$$

p détermine la *régularité* de l'ondelette. Une régularité forte garantit de bonnes propriétés de décorrélation. En particulier, si une fonction 1D f est de classe $\mathbf{C}^\alpha$, $\alpha \leqslant p$, sur un intervalle contenant le support de l'ondelette $\psi_{a,b}$, alors le produit scalaire $< f, \psi_{a,b} >$ va être quasiment nul. Ainsi, plus p est élevé plus la famille d'ondelettes pourra représenter une large classe de régularités.
La localisation. Comme ψ a un support compact, le paramètre d'échelle ouvre l'accès à l'analyse de phénomènes oscillatoires arbitrairement localisés dans le temps. Comme nous l'avons vu, ceci se fait au prix d'une perte de localisation en fréquence : quand a tend vers 0, les ondelettes $\psi_{a,b}$ sont visualisées par des rectangles très fins en temps (de l'ordre de l'échelle a) et très longs en fréquences (de l'ordre de $1/a$).

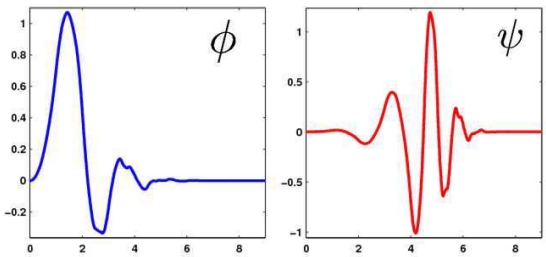


Fig. 1.6 : Fonction d'ondelette ψ de Daubechies [Dau92] à 5 moments nuls et fonction d'échelle ϕ associée.

La base d'ondelettes orthonormées et multi-résolutions est née des travaux de Meyer [Mey88], Daubechies [Dau88] et Mallat [Mal89]. Elle est construite en utilisant un échantillonnage astucieux des échelles et des temps correspondant à une partition dyadique du plan temps-fréquence. Dans le cas de signaux réels, cette base s'écrit :

$$\mathcal{B} = \{\psi_{j,m} \setminus j \geqslant 0, m \in \mathbb{Z}\} \text{ avec } \psi_{j,m} = 2^{-j/2}\psi(2^{-j}t - m) \tag{1.17}$$

Dans le cas discret, l'échelle j et le paramètre de translation m sont limités par la dimension du signal. Supposons que f est un signal discret de dimension 2^N. Sa décomposition en ondelettes est obtenue en calculant les produits scalaires $d_j[m] =< f, \psi_{j,m} >$ appelés *coefficients d'ondelettes*. Pour des raisons pratiques, on préfère en général avoir une décomposition sur un nombre limité d'échelles. Pour une échelle $j \geqslant 0$ donnée, on définit alors les fonctions suivantes :

$$\phi_{j,m}(t) = \sum_{k=0}^{j} \psi_{k,m} \tag{1.18}$$

Ces fonctions peuvent être déterminées par dilatation et translation d'une même fonction ϕ appelée fonction d'échelle. Notons V_1 le sous-espace de $\mathcal{L}^2(\mathbb{R})$ engendré par la famille de fonctions $\{\phi_{1,m}\}_{m\in\mathbb{Z}}$ et W_1 le sous-espace de $\mathcal{L}^2(\mathbb{R})$ engendré par la famille de fonctions $\{\psi_{1,m}\}_{m\in\mathbb{Z}}$. Un niveau de décomposition d'ondelettes consiste à projeter f sur V_1 et W_1. La projection sur V_1 produit une approximation ou basse fréquence de f de dimension 2^{N-1}, tandis que la projection sur W_1 produit une sous-bande de détails de dimension 2^{N-1}. Comme V_1 et W_1 sont orthogonaux, la décomposition est réversible. De même tout espace V_j peut être décomposé en deux espaces orthogonaux V_{j+1} et W_{j+1}. Ceci permet de construire une pyramide multi-résolutions de f en décomposant récursivement la basse fréquence. La propriété **multi-résolutions** de la transformée en ondelettes peut être caractérisée par l'emboîtement des espaces dans $\mathcal{L}^2(\mathbb{R})$:

$$\begin{aligned} V_{j-1} &= V_j \oplus W_j \\ \mathcal{L}^2(\mathbb{R}) &= \oplus_{j\geqslant 0} W_j = V_j \oplus_{k\geqslant j} W_k \end{aligned} \tag{1.19}$$

On obtient ainsi la représentation de f sur un nombre fini d'échelles $J \geqslant 1$, appelé *niveau de décomposition* :

$$f(t) = \sum_{m=0}^{2^{N-J}-1} < f, \phi_{J,m} > \phi^*_{J,m}(t) + \sum_{k=1}^{J} \sum_{m=0}^{2^{N-k}-1} < f, \psi_{k,m} > \psi^*_{k,m}(t) \tag{1.20}$$

où $\phi^*_{j,m}$ $\psi^*_{j,m}$, $j \geqslant 0$, $m \in \mathbb{Z}$, sont les fonctions de synthèse correspondant à $\phi_{j,m}$ et $\psi_{j,m}$. La première partie du membre de droite correspond à l'approximation (basse fréquence) de f à l'échelle 2^J et la seconde à une somme de détails (hautes fréquences).

Optimalité de la base 1D Les bonnes propriétés de la base d'ondelettes en font un outil d'analyse efficace pour des fonctions 1D ayant un nombre fini de discontinuités. En particulier, si la fonction ψ a p moments nuls et que f est $\mathcal{C}^\alpha$ par morceaux, avec $\alpha \leqslant p$, on peut montrer [Pey05b] qu'une telle base aboutit à une décroissance de l'erreur d'approximation du type :

$$\|f - \tilde{f}_M\|^2 \leqslant CM^{-2\alpha} \tag{1.21}$$

qui correspond à la décroissance optimale atteignable pour cette classe de signaux [DeV98]. Comme une ondelette est oscillante, remarquons que le phénomène de Gibbs reste présent lors du filtrage des hautes fréquences.

1.3.5.2 Bases 2D, dD

La base d'ondelettes 2D est construite en effectuant les produits tensoriels des sous-espaces 1D. Ceci revient à réaliser des translations et dilatations de trois ondelettes mères $\left\{\psi^H, \psi^V, \psi^D\right\}$ telles que :

$$\psi^H(x,y) = \psi(x) \otimes \phi(y), \quad \psi^V(x,y) = \phi(x) \otimes \psi(y), \quad \psi^D(x,y) = \psi(x) \otimes \psi(y) \tag{1.22}$$

où ψ est l'ondelette mère 1D et ϕ est la fonction d'échelle 1D (figure 1.7). On parle de bases d'ondelettes *séparables* car le filtrage peut se faire indépendamment dans les deux dimensions horizontale et verticale.

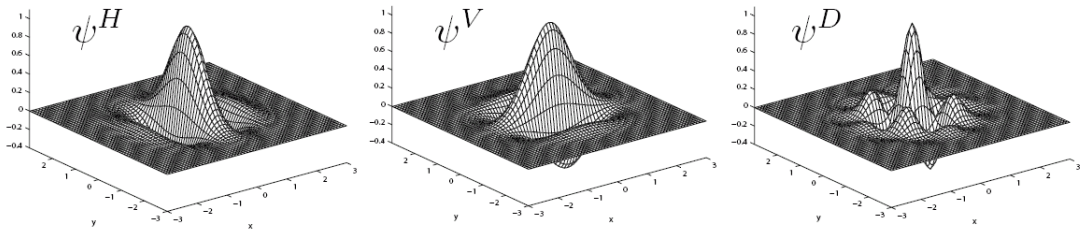


Fig. 1.7 : Un triplet d'ondelettes 2D. D'après [Pey05b].

Comme dans le cas 1D, la base d'ondelettes 2D permet une approximation multi-résolutions d'un signal. A une échelle 2^j donnée, le sous-espace de détails W_j est décomposé en trois sous-espaces $\left\{W_j^H, W_j^V, W_j^D\right\}$ qui permettent respectivement d'isoler des détails globalement verticaux, horizontaux et diagonaux. Des sous-espaces d'approximation V_j peuvent être construits comme dans le cas 1D et correspondent à des dilatations et translations de la fonction d'échelle 2D $\phi(x,y) = \phi(x) \otimes \phi(y)$.

Considérons une image I de dimensions $2^n \times 2^n$. La décomposition de I dans une base d'ondelettes 2D sur $J \geqslant 1$ niveaux s'obtient en calculant les coefficients d'échelle $a_J[\mathbf{m}]$ pour chaque translation $\mathbf{m} \in [0 \dots 2^{n-J} - 1]^2$ et les coefficients d'ondelette $d_j^\theta[\mathbf{m}]$ pour

chaque échelle $j \in \{1 \dots J\}$, orientation $\theta \in \{H, V, D\}$ et translation $\mathbf{m} \in [0 \dots 2^{n-j} - 1]^2$:

$$\begin{aligned} a_J[\mathbf{m}] &=< I, \phi_{J,\mathbf{m}} > \quad \text{avec} \quad \phi_{J,\mathbf{m}} = 2^{-J}\phi(2^{-J}x - m, 2^{-J}y - n) \\ d_j^\theta[\mathbf{m}] &=< I, \psi_{j,\mathbf{m}}^\theta > \quad \text{avec} \quad \psi_{j,\mathbf{m}}^\theta = 2^{-j}\psi^\theta(2^{-j}x - m, 2^{-j}y - n) \end{aligned} \tag{1.23}$$

Notons qu'une ondelette $\psi_{j,\mathbf{m}}^\theta$ est localisée au voisinage du point $2^j\mathbf{m}$. Chaque coefficient peut ainsi être localisé dans le domaine image $\mathcal{D}$. Chaque ensemble de détails d_j^θ peut aussi être considéré comme une image de dimensions $(2^{n-j}-1)\times(2^{n-j}-1)$. Ceci permet une interprétation visuelle directe de la décomposition en ondelettes. La figure 1.8 (a) montre le résultat d'une décomposition en ondelettes de l'image *Lena* sur 5 niveaux.

Fig. 1.8 : (a) Décomposition en ondelettes sur 5 niveaux, (b) Reconstruction en gardant 10% des coefficients de plus grande amplitude, (c) Idem en gardant 3% des coefficients de plus grande amplitude. L'ondelette de Daubechies 9/7 [ABMD92] est utilisée ici.

Notons enfin que la construction d'une base d'ondelettes pour des signaux de dimension d quelconque se fait en suivant le même cheminement que pour le cas 2D.

Succès des ondelettes séparables. Outre les qualités d'approximation des bases d'ondelettes (figure 1.8(b)), leur attrait principal par rapport aux bases de Fourier réside dans leur capacité à représenter une image sur plusieurs niveaux de résolution, modélisant ainsi une caractéristique essentielle de la vision humaine [Fie93]. Lors d'une approximation, il devient même possible d'imiter le phénomène de masquage opéré par l'œil sur les hautes fréquences pour évaluer la qualité du signal. Ceci se fait simplement en attribuant des poids différents à chaque échelle lors du calcul de l'erreur quadratique [Tau99]. Mais la propriété de multi-résolutions spatiale est surtout un atout important dans un contexte de codage « scalable » (voir paragraphe 1.4.3) car elle permet de générer facilement un flux emboîté décodable à différentes résolutions d'affichage. Cette « scalabilité » intrinsèque est exploitée de façon performante par le standard JPEG2000 qui est un des aboutissements majeurs des ondelettes en termes d'applications [TM01]. Il faut néanmoins noter que le succès de ce standard ne repose pas que

sur la brique de la transformée. Il réside aussi dans la capacité de son codeur entropique à exploiter les faiblesses de l'ondelette 2D.

Faiblesses des ondelettes séparables. Les limites des ondelettes apparaissent si l'on étudie leur résultat théorique en termes d'approximation non linéaire. En effet, on peut montrer que si I est une image de classe $\mathcal{C}^\alpha \backslash \mathcal{C}^\alpha$ et que l'ondelette mère a une régularité d'ordre $p \geqslant \alpha$ alors l'erreur quadratique satisfait une décroissance du type [PM05] :

$$\|I - \tilde{I}_M\|^2 \leqslant K \cdot M^{-1} \tag{1.24}$$

On observe donc une amélioration par rapport à la représentation en cosinus discret. Cependant, cette décroissance n'est pas optimale dès que $\alpha > 1$. La sous-optimalité de la base d'ondelettes en 2D s'explique essentiellement par son incapacité à exploiter la géométrie des images. En particulier, on peut mettre en avant deux faiblesses de la représentation en ondelettes.

Manque de variété directionnelle. Pour une ondelette unidimensionnelle de régularité d'ordre p, les ondelettes mères ψ^H et ψ^V sont régulières d'ordre p respectivement le long de la direction horizontale et de la direction verticale *uniquement*. ψ^D est régulière dans les deux directions. A une échelle 2^j, ceci permet d'isoler dans une seule sous-bande une singularité 1D exactement horizontale, verticale ou diagonale. Dans toutes les autres directions, l'ordre de régularité est nul. Ceci limite fortement la capacité de l'ondelette à capturer les régularités le long de contours *courbes* par exemple. A une échelle 2^j, l'énergie d'une singularité 1D de forme quelconque est de ce fait propagée sur les trois sous-bandes $\{H, V, D\}$ (figure 1.8(a)). Ceci a une conséquence visuelle directe lors d'une approximation. En effet, le seuillage d'un coefficient d'ondelette à une échelle 2^j et orientation θ se traduit par l'apparition d'oscillations de Gibbs comme dans le cas 1D. Si l'énergie d'une singularité est répartie à une échelle 2^j sur les trois sous-bandes, alors cette énergie est susceptible d'être tronquée lors d'une approximation et des oscillations peuvent donc apparaître dans les trois directions. Ceci génère l'artefact visuel connu sous le nom de phénomène de « ringing »(figure 1.8(c)).

Ratio d'aspect fixe. Par *ratio d'aspect* nous désignons le rapport entre l'élongation du support dans les directions horizontale et verticale. Si l'ondelette mère ψ et la fonction d'échelle ϕ ont un support de taille respectif m_1 et m_2, alors le ratio d'aspect de ψ^H, ψ^V et ψ^D est respectivement m_1/m_2, m_2/m_1 et 1. A une échelle fine (j proche de 1), un noyau $\psi^\theta_{j,\mathbf{m}}$ ne peut donc capturer des régularités que dans un petit voisinage du point $2^j\mathbf{m}$. Ainsi, un contour quelconque *même horizontal ou vertical* produira un nombre de coefficients signifiants proportionnel à sa longueur dans chaque bande de fréquence (figure 1.9(a)).

Ainsi, nous observons que dans le cas 2D, l'ondelette séparable n'est pas capable d'exploiter efficacement la régularité le long des singularités même les plus simples. Dans le cas d'un signal 3D comme une vidéo, il n'existe pas à notre connaissance de résultat

d'approximation théorique avec une ondelette séparable. Cependant, on comprend qu'un filtrage séparable le long de directions fixes est encore plus pénalisant pour une vidéo car les trajectoires de régularité temporelle dues au mouvement ne sont pas prises en compte.

1.3.6 Nécessité d'exploiter la géométrie et le mouvement : les ondelettes « seconde génération »

Pour améliorer les performances de la représentation en ondelettes, l'exploitation de la géométrie et du mouvement apparaît comme une condition nécessaire. Dans les chapitres 2 et 3, nous présenterons plusieurs outils proposés dans la littérature pour tirer avantage des corrélations spatiales puis temporelles. Le but est de construire des fonctions de base capables de capturer la régularité le long des lignes de flux géométrique et temporel. Conformément à la figure 1.9(b), il faut pour ce faire apporter une dose plus importante d'**anisotropie** aux atomes élémentaires. En particulier, il paraît important que la famille de représentation comporte des atomes orientés selon un nombre « suffisant » de directions dans l'espace et le temps. Pour capturer les régularités sur un voisinage spatio-temporel suffisamment grand, il faut aussi que la famille de représentation comporte des atomes possédant des ratios d'aspect divers. Plus généralement, le but est de créer une représentation à base d'ondelettes de formes variées parfois dites *ondelettes seconde génération*.

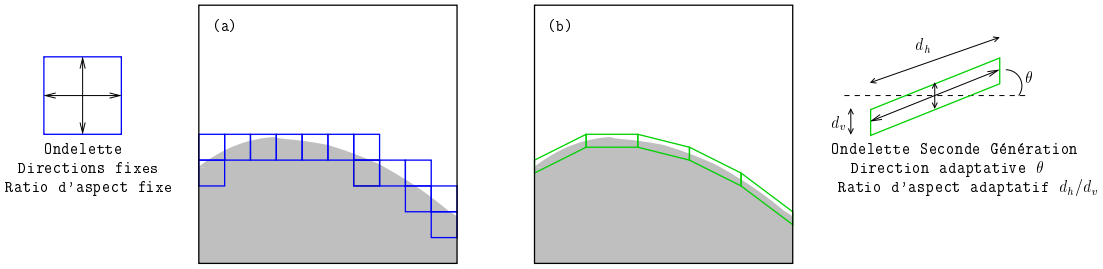


Fig. 1.9 : (a) La décomposition en ondelettes génère une grand nombre de coefficients signifiants autour d'une discontinuité, (b) Intégrer une dose d'anisotropie permet de capturer le contour et de générer un petit nombre de coefficients signifiants.

Si la majorité des ondelettes seconde génération adoptent une approche comme celle illustrée figure 1.9 où le but est d'adapter l'ondelette au contenu de l'image, l'étude que nous proposerons au chapitre 4 adopte une approche inverse : l'idée est d'adapter le contenu de l'image à une transformée en ondelettes séparables. Ces deux approches sont similaires et le but recherché reste le même : concentrer l'énergie de l'image sur un nombre limité de coefficients d'ondelettes.

1.4 Compression

La compression d'images est le cadre applicatif dans lequel nos travaux se situent. Dans cette section, nous rappelons tout d'abord les briques de base constitutives d'un codeur par transformée. Nous revenons ensuite sur le problème de l'allocation de débit avant de nous arrêter sur un enjeu important de la compression : la scalabilité. Enfin, nous faisons un focus sur les codeurs d'images par ondelettes EZW [Sha93], SPIHT [SP96] et EBCOT [Tau99]. Ces codeurs parviennent à compenser les faiblesses des ondelettes en s'appuyant sur une modélisation probabiliste des résidus de corrélation.

1.4.1 Briques de base

Le but d'un algorithme de compression est de minimiser l'espace requis pour stocker ou transmettre une information avec une certaine qualité. Pour ce faire, le codage par transformée s'appuie sur trois briques schématisées sur la figure 1.10 :

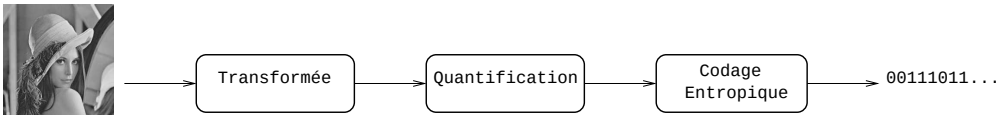


Fig. 1.10 : Briques de base d'une compression par transformée.

Transformée. La brique de transformée s'intéresse à la problématique de représentation, telle que nous l'avons définie à la section précédente. Cette brique n'introduit pas de distorsion. Dans le cas de l'image fixe, le lien entre compression et approximation linéaire est explicité par Mallat et Falzon dans [MF98] : pour des débits inférieurs à 1 bpp, les auteurs montrent en effet que les performances d'un codeur par transformée dépendent de la capacité à approximer une image avec un petit nombre de coefficients non nuls.

Quantification. La brique de quantification permet de contrôler le taux de compression en fixant un pas de quantification plus ou moins élevé. Un codeur est dit sans perte ou quasi sans perte si la quantification ne produit pas de dégradation visuellement perceptible du signal. Lorsque l'espace de stockage ou la bande passante sont limités, il est parfois nécessaire de recourir à un codage avec perte. On distingue généralement les quantificateurs *scalaires* des quantificateurs *vectoriels* [Gra84]. Comme la problématique de quantification n'est pas le point central de cette thèse, nous renvoyons le lecteur aux ouvrages [AR02, RG98, LLo82].

Codage entropique. A l'issue de l'étape précédente, chaque coefficient du signal quantifié est associé à un *symbole* dans un alphabet $\mathcal{A}$. La suite de symboles obtenue est notée $\mathbf{x} = \{x_t \backslash x_t \in \mathcal{A}\}_{t\in\mathbb{N}}$. Considérons ces valeurs comme les réalisations d'un

processus aléatoire discret décrit par la variable aléatoire X (dite *source*). En notant $\mathbf{P}\{X = a_i\}$ la probabilité de voir apparaître le symbole a_i, on définit l'*entropie* de la source X comme :

$$\mathbf{H}(X) = -\sum_{a_i \in \mathcal{A}} \mathbf{P}\{X = a_i\} \log_2 \mathbf{P}\{X = a_i\} \tag{1.25}$$

L'entropie mesure la quantité moyenne d'information par symbole. Elle s'exprime en bits/symbole. D'après la théorie de l'information proposée par Shannon [Sha48], les réalisations du processus aléatoire X ne peuvent être représentées sans erreur en un nombre moyen de bits inférieur à cette quantité. Le codage entropique permet d'atteindre cette borne inférieure en utilisant par exemple un codage de Huffman [Huf52] ou un codage arithmétique [WNC87, Sai03]. Supposons maintenant que la suite de symboles $\mathbf{x}$ puisse être décomposée en deux suites $\mathbf{x}_1$ et $\mathbf{x}_2$ réalisations de deux processus aléatoires X_1 et X_2. On définit alors l'*entropie jointe* des deux sources X_1 et X_2 comme :

$$\mathbf{H}(X_1, X_2) = -\sum_{a_i \in \mathcal{A}} \sum_{b_i \in \mathcal{A}} \mathbf{P}\{X_1 = a_i, X_2 = b_i\} \log_2 \mathbf{P}\{X_1 = a_i, X_2 = b_i\} \tag{1.26}$$

On montre que l'entropie jointe vérifie la propriété suivante :

$$\mathbf{H}(X_1, X_2) \leqslant \mathbf{H}(X_1) + \mathbf{H}(X_2) \tag{1.27}$$

avec égalité uniquement lorsque les variables sont indépendantes. Le plus souvent, la décorrélation effectuée lors de la transformée n'est pas parfaite et donc des résidus de corrélation subsistent entre certains groupes de coefficients. Pour tendre vers l'entropie jointe, les codeurs utilisent donc des *contextes* qui permettent de prédire les réalisations de X_2 sachant les réalisations de X_1. Ceci revient à réécrire l'entropie jointe comme :

$$\mathbf{H}(X_1, X_2) = \mathbf{H}(X_1) + \mathbf{H}(X_2|X_1) \tag{1.28}$$

et à trouver les contextes permettant de tendre vers l'*entropie conditionnelle* $\mathbf{H}(X_2|X_1)$. Bien sûr, l'idéal serait de trouver une transformée qui réduise au maximum l'utilité de ces contextes.

D'après ce bref descriptif des briques élémentaires, nous remarquons que la problématique de compression est avant tout une problématique de représentation. Mais nous remarquons également que la brique de codage peut jouer un rôle important dans la recherche des résidus de corrélations. De ce fait, le taux de décroissance d'une approximation non linéaire ne peut résumer à lui seul le potentiel d'une représentation dans une application de compression. Certaines représentations peuvent donner des résidus de corrélation faciles à exploiter. D'autres peuvent donner de meilleurs résultats en termes d'approximation non linéaire mais aboutir à des résidus plus difficiles à coder. D'une manière générale, le but d'un algorithme de compression est de trouver un exposant α le plus grand possible tel que :

$$\mathbf{D}(\mathbf{R}) \leqslant K \cdot \mathbf{R}^{-\alpha} \tag{1.29}$$

où K est une constante qui ne dépend que du signal. $\mathbf{R}$ est le débit occupé par le flux généré en sortie du codeur entropique et $\mathbf{D}$ est idéalement une mesure de la distorsion *visuelle* introduite par la quantification des coefficients. L'exposant α peut donc différer de celui obtenu par l'étude de l'approximation non linéaire selon la capacité du codeur à exploiter les résidus de corrélation. Notons enfin que lorsque le flux binaire généré est destiné à être transmis par un *canal*, la qualité de l'image décodée en bout de chaîne dépend aussi des bruits de transmission et donc de la capacité à *sécuriser* les données transmises. Même si cette thèse n'est pas dédiée au codage conjoint source-canal, il est important de garder ceci en tête lors de l'évaluation d'un algorithme. Un panorama du codage conjoint source-canal peut être trouvé dans [ADR96].

1.4.2 Optimisation débit-distorsion

Toute la difficulté des algorithmes de compression est de trouver un compromis idéal entre le *débit* $\mathbf{R}$, occupé par le flux binaire en sortie du codeur, et la *distorsion* $\mathbf{D}$ du signal reconstruit au décodage. Dans ce paragraphe, nous supposons que le débit et la distorsion dépendent d'un jeu de paramètres Θ, par exemple un ensemble de pas de quantification associé à une certaine partition des coefficients. Nous nous arrêtons ici sur les principes d'une optimisation Lagrangienne car elle est très largement utilisée dans les algorithmes de compression. Nous verrons notamment au chapitre 2 que certaines méthodes s'appuient sur une telle optimisation pour estimer des paramètres géométriques.

L'optimisation Lagrangienne est expliquée en détails dans [Ram93b]. Le but d'une telle optimisation est de trouver le jeu de paramètres optimal $\Theta^\star$ qui minimise la distorsion moyenne $\mathbf{D}$ sous la contrainte d'un débit cible $\mathbf{R}_{cible}$:

$$\Theta^\star = \arg\min_{\Theta} \mathbf{D}(\Theta) \quad \backslash \quad \mathbf{R}(\Theta) \approx \mathbf{R}_{cible} \tag{1.30}$$

On montre [Ram93b] que ce problème d'optimisation sous contrainte est équivalent à un problème non contraint décrit par :

$$\Theta^\star = \arg\min_{\Theta} \mathbf{J}(\lambda) = \arg\min_{\Theta} \mathbf{D}(\Theta) + \lambda \mathbf{R}(\Theta) \quad \backslash \quad \mathbf{R}(\Theta) \approx \mathbf{R}_{cible} \tag{1.31}$$

Le débit $\mathbf{R}$ et la distorsion $\mathbf{D}$ sont maintenant incorporés au coût Lagrangien $\mathbf{J}$ pour un multiplicateur $\lambda \geqslant 0$ donné. Le multiplicateur de Lagrange dicte le compromis entre débit et distorsion. L'ensemble des points débit-distorsion $(\mathbf{R}, \mathbf{D})$ atteignables peut être placé dans un espace 2D. Avec une optimisation Lagrangienne, seuls les points situés sur une courbe dite *courbe opérationnelle* peuvent être atteints (voir figure 1.11). Pour un débit cible donné, le multiplicateur optimal $\lambda^\star$ est la pente de la courbe opérationnelle en ce débit. En général, ce multiplicateur n'est pas connu *a priori*. Cependant, si la

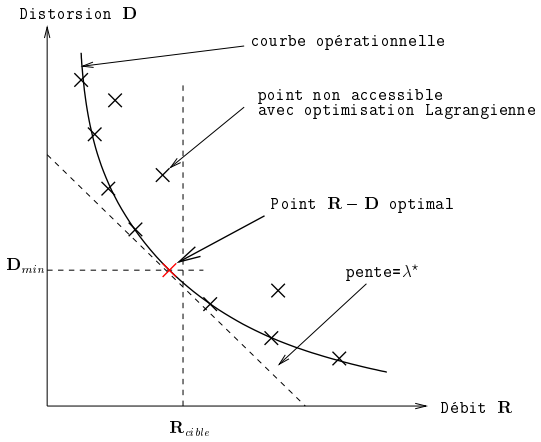


Fig. 1.11 : Courbe débit-distorsion caractéristique d'un système de compression.

courbe opérationnelle est supposée convexe alors les paramètres optimaux $\Theta^\star$ peuvent être trouvés par exemple par une recherche dichotomique (voir [Ram93b]) .

Comme nous le verrons dans la suite, les coefficients d'une image sont souvent traités (transformés, quantifiés, codés) par groupe. C'est le cas par exemple dans les normes JPEG [Wal91] et JPEG2000 [SCE01] où les coefficients sont codés bloc par bloc. A chaque bloc correspond alors un ensemble de paramètres et la concaténation de tous ces ensembles donne l'ensemble Θ. La complexité de l'optimisation augmente donc d'un degré. Néanmoins, lorsque la transformée utilisée est orthonormale, l'optimisation peut se faire indépendamment sur chaque bloc. Supposons que les coefficients soient regroupés en blocs b_i. Dans chaque bloc, un certain choix de paramètres Θ_i aboutit à une distorsion $\mathbf{D}_i$ et un débit $\mathbf{R}_i$. Si la base de représentation est orthonormale, on montre alors que la distorsion totale de l'image et le débit total du flux binaire généré s'expriment comme la somme des distorsions et débits sur chaque bloc :

$$\begin{aligned} \mathbf{D}(\Theta) &= \sum_i \mathbf{D}_i(\Theta_i) \\ \mathbf{R}(\Theta) &= \sum_i \mathbf{R}_i(\Theta_i) \end{aligned} \tag{1.32}$$

Ce problème d'*allocation de ressources indépendantes* peut être résolu en recherchant pour chaque bloc b_i le couple ($\mathbf{R}_i$,$\mathbf{D}_i$) correspondant au multiplicateur optimal $\lambda^\star$.

1.4.3 Objectif « scalabilité »

Dans les paragraphes précédents, certains enjeux fondamentaux liés à la compression ont été mis en avant. Depuis quelques années, la *scalabilité* est apparue comme un nouvel enjeu fort [Sha93, SP96, Tau99, JVT06]. L'objectif est de créer un flux binaire *emboîté* qui puisse être tronqué selon des contraintes liées à la chaîne de transmission ou au

dispositif d'affichage disponible au décodage. Typiquement, un flux emboîté comporte une *couche de base* permettant de satisfaire des contraintes maximales, et un ensemble de *couches de raffinement* permettant de s'adapter à des contraintes de moins en moins restrictives. Ceci permet de n'encoder qu'un seul flux pour différents utilisateurs. Quatre types différents de scalabilité peuvent être mis en avant [Cam04b] :

Scalabilité SNR ou scalabilité en qualité. La scalabilité en qualité est la possibilité de réduire la distorsion de quantification entre le signal original et le signal reconstruit. A chaque niveau de hiérarchie est associée une qualité de reconstruction. La couche de base permet de reconstruire le signal avec une qualité minimale. L'ajout d'information supplémentaire permet d'améliorer cette qualité.

Scalabilité spatiale. Les dispositifs d'affichage utilisés par les clients peuvent avoir des caractéristiques variées. Notamment, plusieurs résolutions d'affichage existent selon que le terminal est un téléphone portable, un assitant personnel PDA (« Personal Digital Assistant »), un écran TV, SD-TV (« Standard Definition Television »), HD-TV (« High Definition Television »)... La *scalabilité spatiale* est donc la capacité à générer un flux emboîté dont chaque couche regroupe les informations spécifiques à une résolution d'affichage.

Scalabilité temporelle. Dans le cas d'une vidéo, une autre caractéristique variable des terminaux est la fréquence d'affichage (en Hz). La *scalabilité temporelle* est la capacité à ajouter des images intermédiaires entre les images reconstruites par la couche de base.

Scalabilité en complexité. La scalabilité en complexité est nécessaire lorsqu'on souhaite implémenter un même algorithme sur des dispositifs embarqués aux fortes contraintes matérielles et sur des dispositifs moins contraints comme des récepteurs de télévision numérique.

Notons que la propriété scalable d'un codeur est un avantage important pour pouvoir réagir aux aléas de la chaîne de transmission et ainsi assurer une *qualité de service* (QoS) minimale au plus grand nombre de clients. Ainsi, la troncature de certaines couches du flux peut être utilisée comme solution de repli lorsqu'un réseau se surcharge. Notons enfin que si l'objectif est de créer un flux complètement scalable, alors toutes les informations représentant le signal doivent pouvoir être encodées de façon scalable. Cette remarque est importante si l'on souhaite extraire un modèle de géométrie ou de mouvement dans des images. Nous y reviendrons dans les chapitres 2 et 3.

1.4.4 Codeurs ondelettes

Dans la section précédente, nous avons noté que la représentation en ondelettes ne donnait pas un résultat d'approximation optimal dans le cas 2D. En pratique, ceci se traduit par des résidus de corrélation dans les sous-bandes de détails. En observant

l'amplitude des détails (voir figure 1.8(a)), on voit néanmoins que ces résidus sont des résidus de *géométrie* et qu'il est possible de les caractériser. Pour une orientation donnée, on voit par exemple que les détails sont corrélés entre les échelles. Pour une orientation et échelle données, on voit aussi que des corrélations spatiales subsistent. Une étude des dépendances inter-échelles et intra-échelle est proposée par Liu et Moulin [LM01]. La caractérisation de ces dépendances a permis d'aboutir à des codeurs ondelettes efficaces. Nous revenons brièvement sur la façon dont sont exploités les résidus de corrélation dans ces codeurs.

Codage inter sous-bandes. Le codeur par arbres de zéros EZW de Shapiro [Sha93] est un des premiers codeurs ondelettes. L'hypothèse principale à la base de ce codeur est la suivante : si un coefficient d'ondelette à une certaine échelle est non signifiant pour un seuil T donné, alors tous les coefficients aux échelles plus fines ayant la même localisation spatiale dans les sous-bandes de même orientation ont une forte probabilité d'être non signifiants pour T [1]. Cette hypothèse, si elle est vérifiée, permet de coder l'ensemble des coefficients d'un arbre (tel que représenté figure 1.12) à l'aide d'un seul symbole. L'arbre est alors dit *arbre de zéros* car tous ses coefficients sont insignifiants par rapport au seuil T courant. Le codeur EZW opère un codage itératif en plans de bits en utilisant des pas de quantification dyadiques de type $T_i = T_{i-1}/2$ et en mettant à jour des listes de *signifiance* et de *raffinement*.

L'algorithme SPIHT (« Set Partitioning in Hierarchical Trees ») proposé par Said et Pearlman [SP96] apporte différentes améliorations à l'algorithme EZW. SPIHT considère les coefficients par groupes. La modification majeure par rapport à EZW réside dans la mise à jour d'une troisième liste permettant de créer des *ensembles non signifiants* de grandes tailles. Ces ensembles non signifiants permettent de connaître l'état d'une descendance même si le coefficient n'est pas la racine d'un zerotree. Leur création nécessite l'utilisation de contextes plus complexes mais permet un codage plus efficace de l'information non signifiante. L'efficacité de cet algorithme a fait de SPIHT une référence en transmission progressive d'images fixes.

Codage intra sous-bandes. Si les codeurs précédents exploitent les dépendances inter-échelles, le codeur EBCOT (« Embedded Block Coding with Optimized Truncation ») introduit par Taubman [Tau99] exploite les dépendances à l'intérieur de chaque sous-bande. A une échelle j et orientation θ, les coefficients d'ondelettes sont ainsi partitionnés en blocs, typiquement de taille 64×64. A l'intérieur de chaque bloc, les coefficients sont découpés en plans de bits en utilisant des pas de quantification dyadiques comme précédemment. Une fois ce découpage effectué, le codage peut commencer. Un parcours particulier des plans de bits des coefficients sous forme de colonne de 4 bits est mis en place. Différents contextes, déterminés empiriquement, permettent de coder efficacement un plan de bits courant en fonction des configurations des bits voisins dans le plan de bits précédent. En vue d'une transmission progressive, le flux binaire généré présente de nombreux points d'arrêt potentiels. Pour chaque bloc, le point d'arrêt

[1] Un détail d est dit *insignifiant* pour un seuil T si $|d| < T$. Il est dit *signifiant* dans le cas contraire.

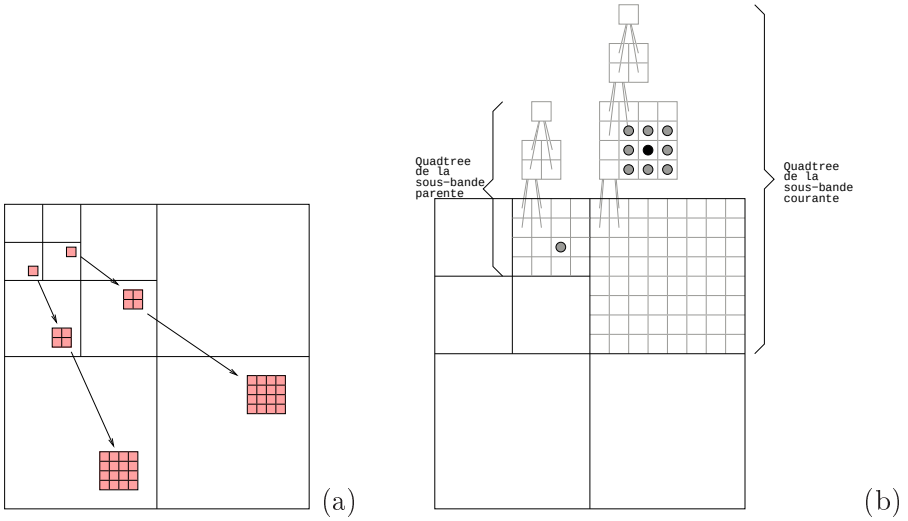


Fig. 1.12 : (a) Modèle de dépendance inter-échelle dans EZW [Sha93] et SPIHT [SP96], (b) Contexte pour le codage de la signifiance dans EZBC : le nœud courant est représenté en noir et ses contextes inter et intra sont représentés en gris.

qui optimise le critère débit-distorsion sur l'image entière est ensuite déterminé. C'est un problème d'allocation de ressources indépendantes qui se résout en recherchant le multiplicateur de langrange optimal $\lambda^\star$. L'algorithme EBCOT offre des performances en compression meilleures que EZW et SPIHT (voir l'étude de Chappelier [Cha05b]). En outre l'implémentation de Taubman [Tau99] permet de générer un flux progressif à granularité fine contenant jusqu'à 50 points de troncature. Même dans ce mode, les performances de EBCOT restent meilleures que celles des codeurs précédents. Précisons que EBCOT est le codeur utilisé dans JPEG2000 [TM01]. Les figures 1.13 et 1.14(a,b) donnent une comparaison numérique et visuelle de JPEG et JPEG2000. Elles prouvent s'il est besoin que la représentation en ondelettes apporte un gain substantiel par rapport à la représentation de Fourier.

Codage intra et inter sous-bandes. Dans le cadre de la compression vidéo, le codeur le plus utilisé à l'heure actuelle est le codeur EZBC (« Embedded Zero Block Coder ») [HW01a]. Ce codeur exploite à la fois les dépendances intra et inter sous-bandes. Pour ce faire, une pyramide multi-résolutions de type Quadtree est calculée pour chaque sous-bande (voir figure 1.12(b)). Aux nœuds du niveau le plus bas du Quadtree sont associées les amplitudes des coefficients de la sous-bande en question. Au niveau supérieur, la valeur d'un nœud est la valeur maximale des 4 nœuds fils au niveau inférieur. La racine du Quadtree est donc l'amplitude maximale de la sous-bande. Les différents niveaux de l'arbre sont codés de la racine aux feuilles en testant la signifiance des nœuds. Pour chaque nœud, un contexte est formé à partir de l'état de signifiance de ses huit voisins situés au même niveau de l'arbre, ainsi que du nœud situé dans l'arbre de la sous-bande parente au niveau inférieur. Le fait de considérer le nœud de la

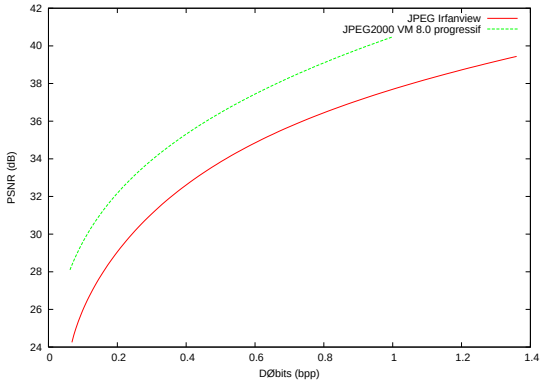


Fig. 1.13 : Comparaison des courbes PSNR=f(Débit) de JPEG et JPEG2000 avec l'image *Lena* 512. La compression JPEG a été effectuée avec le logiciel Irfanview. La compression JPEG2000 a été effectuée avec le codeur VM (« Verification Model ») 8.0 en utilisant le mode progressif à granularité fine.

sous-bande parente au niveau inférieur permet de tenir compte du changement d'échelle entre les sous-bandes et d'exploiter la dépendance inter échelles. Comme EBCOT, EZBC permet de générer un flux progressif. Les performances de ces deux codeurs sont très proches [Cha05b].

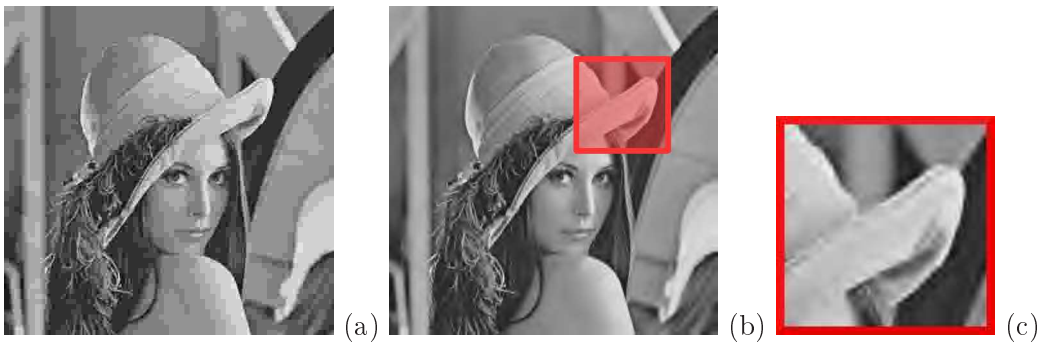


Fig. 1.14 : (a) Image décodée avec JPEG à $0,11$bpp (PSNR $= 27,33$dB), (b) avec JPEG2000 à $0,1$bpp (PSNR $= 29,80$dB), (c) zoom illustrant la limite de JPEG2000.

Ainsi, les algorithmes décrits ci-dessus démontrent qu'il est possible d'exploiter la structure des résidus de corrélation dans le domaine ondelettes pour aboutir à des codeurs performants. Cependant, les modèles statistiques utilisés ne prennent pas réellement en compte les critères géométriques tels que la directionnalité ou l'élongation d'un contour. L'optimisation se fait toujours à l'échelle du pixel et les statistiques des contextes sont apprises à la volée. Plus on se situe bas en débit, plus les limites de la représentation en ondelettes ont un impact sur le résultat visuel de l'image décodée : des rebonds ou effets de pixellisation apparaissent près des contours (figure 1.14(c)). Si

l'on veut améliorer la qualité visuelle d'une image pour un débit donné, il semble donc important d'intégrer les critères géométriques dans la représentation, comme indiqué au paragraphe 1.3.6. Précisons que les algorithmes SPIHT, EBCOT et EZBC peuvent être étendus au codage de sous-bandes ondelettes spatio-temporelles dans le cas d'une vidéo [KXP00, XXLZ01, HW01b]. Cependant ces codeurs s'appuient sur une modélisation du mouvement et sont donc déjà *adaptatifs*.

Dans le chapitre suivant, nous présentons différents outils antérieurs permettant de modéliser la géométrie d'une image et de modifier la représentation ondelettes en conséquence. Nous nous pencherons au chapitre 3 sur les outils antérieurs permettant de même la modélisation et l'exploitation du mouvement dans une séquence vidéo. Dans les chapitres 4 et 5 nous proposerons ensuite nos solutions au problème d'adaptivité pour l'image fixe puis pour la vidéo.

Chapitre 2

Adaptivité spatiale dans les codeurs d'images : outils antérieurs

Comme nous l'avons vu au chapitre précédent, l'ondelette 2D standard est un outil de représentation puissant mais qui ne prend pas en compte les régularités géométriques d'une image. La modélisation probabiliste des résidus de corrélation (voir paragraphe 1.4.4) est une manière de compenser ce défaut de l'ondelette. Néanmoins, elle n'est pas complètement satisfaisante car elle ne résout pas la problématique de représentation. Les méthodes que nous décrivons dans ce chapitre se concentrent sur cette problématique. Elles reposent toutes sur une modélisation de la géométrie, implicite ou explicite, qui dicte la forme et la direction du support de représentation. Elles sont communément divisées en deux catégories.

Les méthodes non adaptatives reposent sur des familles de noyaux de représentation fixes possédant des formes variées mais indépendantes de l'image à analyser. La modélisation géométrique est donc implicite et aucune information annexe n'est nécessaire pour représenter une image.

Les méthodes adaptatives proposent une modélisation explicite de la géométrie. Une géométrie est extraite de l'image et les noyaux de représentation sont formés en fonction de cette géométrie. A chaque image correspondent donc des noyaux différents. Dans un contexte de codage, ceci signifie que l'information de géométrie doit être codée et transmise en plus des coefficients transformés pour pouvoir construire les noyaux de synthèse au décodage. On voit donc apparaître un compromis entre une adaptivité forte, à savoir une représentation précise de la géométrie avec un grand nombre de paramètres, et un faible surcoût de codage avec une représentation moins fine de la géométrie.

Dans la première section de ce chapitre, nous présentons les méthodes non adaptatives les plus connues. Les deux sections suivantes sont consacrées aux méthodes adaptatives. La section 2.2 s'intéresse à des modélisations locales de la géométrie permettant de modifier le noyau d'ondelette séparable. La section 2.3 se concentre sur les

modélisations globales de la géométrie sur l'ensemble du domaine image, avec un focus particulier sur les maillages 2D. La dernière section porte sur le codage des sous-bandes et s'interroge sur les capacités des nouvelles transformées en termes de « scalabilité ».

2.1 Bases fixes

L'ondelette manque d'attributs géométriques. L'objectif ici est de définir de tels attributs pour construire une base qui est *fixe* comme la base d'ondelettes mais pourtant capable de décrire des caractéristiques géométriques diverses. La série de travaux réalisés par Candès et Donoho [CD99b, Don00, CD99a] s'attelle à ce défi ambitieux. Ces travaux se concentrent initialement sur les fonctions définies sur l'espace continu $\mathbb{R}^2$. Étape par étape, ils ont introduit les outils qui ont mené à la construction de la base de Curvelets.

2.1.1 Transformée de Radon

La transformée de Radon [Rad17] d'une fonction I bidimensionnelle est la formulation mathématique d'une projection 1D dans une direction donnée par un angle $\theta \in [0, 2\pi]$ (voir figure 2.1). Elle est définie de la façon suivante :

$$\mathrm{Rad}[I](\theta, t) = \int_{\mathbb{R}} \int_{\mathbb{R}} I(x, y)\delta(x\cos\theta + y\sin\theta - t)\mathrm{d}x\mathrm{d}y$$

où t parcourt toutes les lignes d'orientation θ dans l'espace continu et $\delta(x)$ est l'impulsion de Dirac. L'analyse fréquentielle de la fonction $\mathrm{Rad}[I](\theta, t)$ pour θ fixé correspond à une coupe radiale dans le spectre de I. Il est donc possible de reconstruire la fonction I à l'aide de la totalité des projections $\mathrm{Rad}[I](\theta, t)$. Cette propriété fait de la transformée de Radon un outil privilégié dans les problèmes de reconstruction tomographique où il s'agit de reconstruire une coupe 2D d'un objet 3D à partir de multiple projections réalisées par un scanner. Mais revenons à notre problème de représentation.

Candès et Donoho font l'observation suivante : si I est traversée par une singularité 1D rectiligne orientée de θ, alors la projection de Radon de I dans cette direction réduit la singularité 1D en une singularité de type point. Or, nous avons vu que les ondelettes permettent d'isoler efficacement de telles singularités. Ceci amène naturellement à la construction des Ridgelets.

2.1.2 Ridgelets

Considérons donc la transformée en ondelettes de chaque projection $\mathrm{Rad}[I](\theta, t)$ pour $\theta \in [0, 2\pi]$. Elle est notée $\mathrm{Rid}[I](a, b, \theta)$:

$$\mathrm{Rid}[I](a, b, \theta) = \int_{\mathbb{R}} \psi_{a,b}(t)\mathrm{Rad}[I](\theta, t)\mathrm{d}t$$

où $\psi_{a,b}$ est une ondelette 1D dilatée du facteur d'échelle a et translatée au point $t = b$. Cette transformée peut s'écrire sous une autre forme :

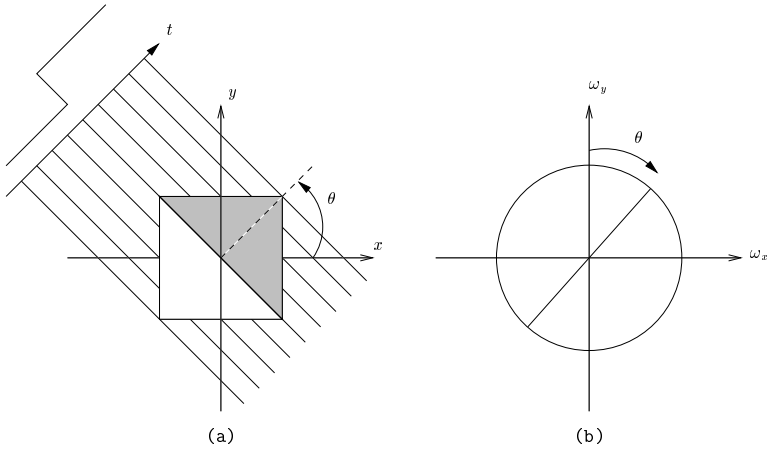


Fig. 2.1 : Coupe de Radon. (a) La projection 1D dans la direction θ transforme une discontinuité de type ligne en une discontinuité de type point, (b) Les coefficients de Fourier de cette projection peuvent être obtenus en effectuant une coupe radiale dans le spectre de I.

$$\mathrm{Rid}[I](a,b,\theta) = \int_{\mathbb{R}}\int_{\mathbb{R}} \psi_{a,b,\theta}(x,y)I(x,y)\mathrm{d}x\mathrm{d}y$$

qui fait apparaître un nouvel atome de représentation $\psi_{a,b,\theta}$: la Ridgelet. Cette Ridgelet est définie à partir de l'ondelette ψ :

$$\psi_{a,b,\theta}(x,y) = a^{-1/2}\psi((x\cos\theta + y\sin\theta - b)/a)$$

La Ridgelet est donc une fonction orientée selon la direction θ et constante le long des lignes $x\cos\theta + y\sin\theta$ (voir figure 2.2). Le facteur d'échelle a caractérise la localisation spatiale de la Ridgelet dans la direction orthogonale à θ. Notons que les coefficients $\mathrm{Rid}[I](a,b,\theta)$ pour un θ fixé représentent une transformée en ondelettes de la projection de Radon. De ce fait, ils correspondent également à une coupe radiale dans le spectre fréquentiel de la fonction I. Dans le domaine spatial, on voit qu'une Ridgelet n'est localisée que dans la direction orthogonale à θ et permet donc de représenter uniquement des singularités rectilignes traversant l'image de part en part. Pour pouvoir représenter des singularités d'ordre plus élevé, il est nécessaire d'apporter une meilleure localisation spatiale dans la direction θ.

La construction des Ortho-Ridgelets (Orthonormal Ridgelets) proposée par Donoho [Don00] tente de répondre à cet objectif. L'idée est de fenêtrer la transformée en Ridgelets pour lui apporter une meilleure localisation. L'auteur considère ainsi une fenêtre de la taille de l'image, mais aussi toutes les fenêtres définies sur des blocs dyadiques de l'image (voir définition d'un bloc dyadique au paragraphe 2.3.1). Cependant, la décomposition de l'image sur cette famille de Ridgelets fenêtrées n'est pas exploitable en pratique pour des signaux numériques car sa discrétisation ne forme pas une frame. La famille d'Ortho-Ridgelets est en quelque sorte trop riche. Candès et Donoho

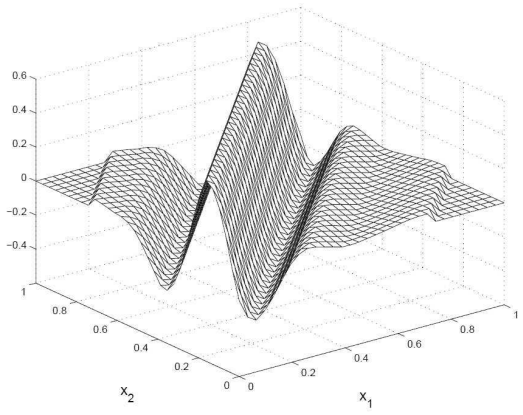


Fig. 2.2 : Une Ridgelet. D'après [Do01b].

proposent alors une nouvelle transformée multi-échelles basée sur un partitionnement particulier du domaine fréquentiel : la transformée en Curvelets.

2.1.3 Curvelets

Pour obtenir une représentation multi-échelles, l'idée est d'effectuer la transformée en Ridgelets sur différentes bandes de fréquences. Candès et Donoho [CD99a] proposent ainsi de découper l'espace des fréquences en couronnes dyadiques $|\omega| \in [2^j, 2^{j+1}]$ et de faire une analyse en Ridgelets de chaque sous-bande. Ils proposent également de discrétiser le paramètre θ de la transformée en Ridgelets en fonction de l'échelle de la sous-bande. On comprend ce choix car utiliser le même nombre de directions pour toutes les échelles n'est pas économique (voir figure 2.3(a)) : pour représenter une sous-bande haute fréquence, il faut utiliser un grand nombre de directions de filtrage. Utiliser la même discrétisation de θ dans une sous-bande de plus basse fréquence introduit de la redondance. Les auteurs proposent alternativement que le nombre d'orientations utilisé pour décrire une sous-bande $[2^j, 2^{j+1}]$ soit le double de celui utilisé pour décrire la sous-bande $[2^{j-2}, 2^{j-1}]$ (figure 2.3(b)). Ils démontrent qu'une telle partition fréquentielle est particulièrement adaptée à la représentation de courbes $\mathcal{C}^2$.

En pratique, le découpage fréquentiel en couronnes n'est pas naturel. La décomposition en Curvelets d'une fonction est alors déclinée en trois temps :

1. Décomposition de l'image en sous-bandes,
2. Fenêtrage lisse de chaque sous-bande en blocs de taille appropriée à chaque échelle,
3. Application sur chaque bloc de la transformée en Ridgelets.

L'objet du fenêtrage des sous-bandes est de donner aux noyaux de représentation un ratio d'aspect particulier adapté aux courbes $\mathcal{C}^2$ (voir figure 2.4). Ce ratio est satisfait en coordonnant la taille de la fenêtre et l'échelle de la sous-bande.

En termes d'approximation non linéaire, la transformée en Curvelets suit la loi :

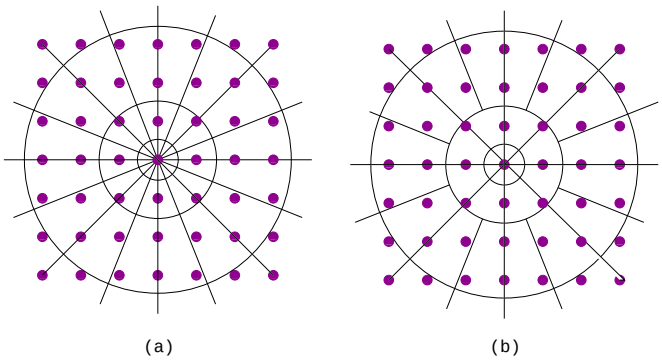


Fig. 2.3 : Discrétisation de la direction θ. (a) Indépendamment de l'échelle et (b) En augmentant le nombre de directions dans les échelles fines (hautes fréquences).

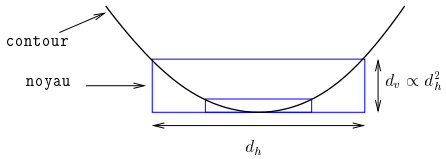


Fig. 2.4 : Ratio d'aspect adapté à un contour $\mathcal{C}^2$. La largeur du support de l'ondelette est proportionnelle au carré de sa longueur.

$$\left\| I - \tilde{I}_M \right\|_2^2 \leqslant CM^{-2}(\log M^3), \qquad M \to \infty$$

Elle est donc quasi-optimale pour les fonctions possédant des singularités C^2. Candès et Donoho ont ainsi démontré qu'il était possible d'obtenir un résultat quasi-optimal avec une base fixe. Ceci les amène à s'interroger sur l'utilité des méthodes adaptatives. Cependant, deux remarques s'imposent. D'une part, on notera que lorsque les singularités ont un ordre de régularité supérieur à 2, la représentation en Curvelets n'est plus optimale. D'autre part, nous répétons que les travaux de Candès et Donoho se concentrent sur les fonctions continues et définissent une partition de l'espace fréquentiel qui n'est pas aisément transposable au cas discret. L'approche en 3 étapes décrites plus haut pose différents problèmes en pratique. En particulier, la transformée en Ridgelets est définie en coordonnées polaires ce qui rend son implémentation problématique si l'on veut limiter la redondance de la représentation. En outre, une telle approche basée blocs nécessite de recourir à des fenêtres qui se chevauchent pour limiter les effets de blocs, et ceci accroît la redondance. L'implémentation discrète de la transformée en Curvelets décrite dans [SCD02] donne ainsi un facteur de redondance égal à $16J + 1$, où J est le nombre d'échelles. Son application à la compression est donc limitée. La question qui se pose est de savoir s'il est possible de construire une transformée à l'aide d'outils discrets et imitant au mieux la décomposition en Curvelets. Dans son travail de thèse [Do01b], Do se penche sur ce problème et aboutit à la transformée en Contourlets.

2.1.4 Contourlets

Les travaux de Do [Do01b] suivent le même cheminement que ceux de Candès et Donoho mais avec une réflexion centrée sur le domaine discret. En se basant sur la théorie des lattices (voir paragraphe 2.2.1) et en se concentrant sur des images carrées dont la dimension est un nombre premier, Do construit tout d'abord une transformée de Radon discrète à reconstruction parfaite et peu redondante. Ensuite, il montre qu'il est possible de construire une transformée orthonormée en Ridgelets en décomposant chaque projection de Radon dans une base d'ondelettes discrète [DV03b]. La transformée obtenue est non-redondante mais suppose que la dimension de l'image est un nombre premier.

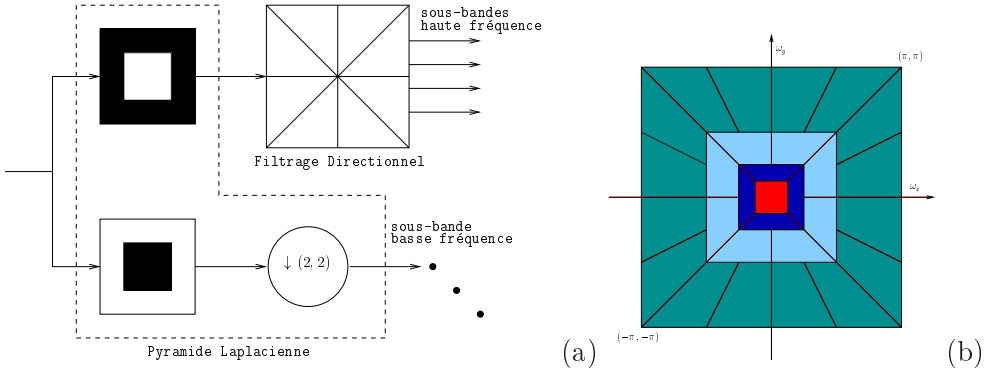
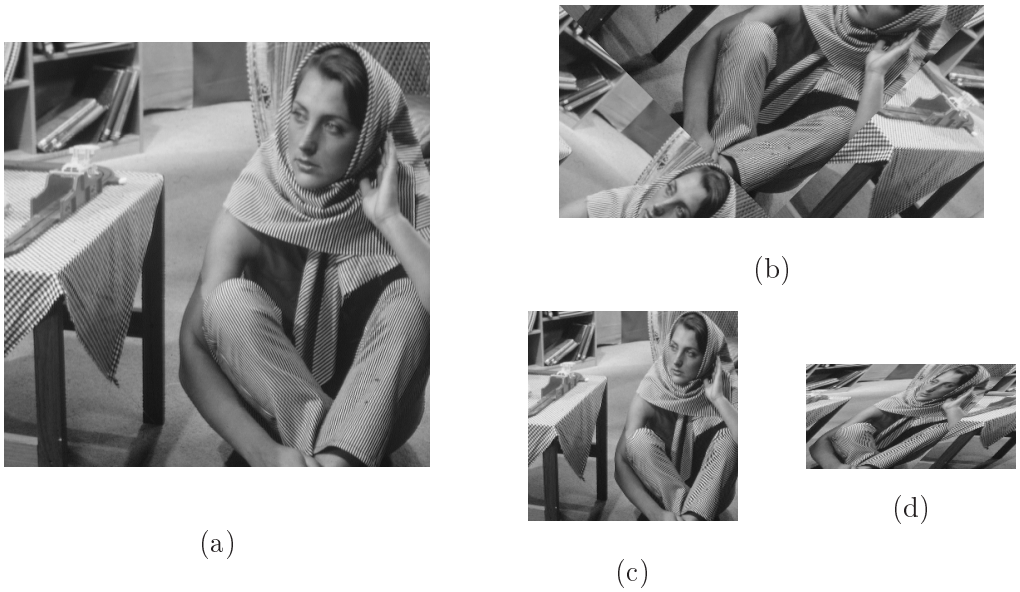


Fig. 2.5 : Principes de la transformée en Contourlets. (a) Combinaison Pyramide Laplacienne et filtrage directionnel, (b) Partition du domaine fréquentiel obtenu.

La dernière étape est la construction d'une transformée discrète multi-échelles, orientée et non adaptative : la transformée en Contourlets [DV05]. Cette transformée est réalisée en combinant analyse multi-résolutions et bancs de filtres directionnels (figure 2.5 (a)). Pour effectuer l'analyse multi-résolutions, les auteurs choisissent d'utiliser la pyramide Laplacienne introduite par Burt et Adelson [BA83]. Au premier niveau, la décomposition pyramidale d'une image I génère deux composantes : une basse fréquence sous-échantillonnée d'un facteur 2 dans les deux dimensions et une haute fréquence non sous-échantillonnée. La haute fréquence est simplement obtenue en soustrayant à I son approximation par la basse fréquence. Le procédé peut être répété récursivement sur la basse fréquence pour obtenir la pyramide. En pratique, les auteurs utilisent une décomposition en ondelettes 9/7 [ABMD92] pour obtenir la basse fréquence. Si une telle décomposition est redondante, le fait de n'avoir qu'une seule sous-bande par échelle facilite l'analyse directionnelle car une singularité ne se trouve pas propagée dans plusieurs sous-bandes.

Une fois la pyramide construite, un banc de filtres directionnels est appliqué sur chaque sous-bande haute fréquence générée. Décrivons la démarche suivie pour une sous-bande donnée de taille $2^n \times 2^n$. Une première passe de filtrage est effectuée dans les directions horizontale et verticale par une paire de filtres dits en éventail. Ceci permet



(a) (b) (c) (d)

Fig. 2.6 : (a) Image *Barbara* d'origine, (b) Premier sous-échantillonnage quinconce, (c) Second sous-échantillonnage quinconce, (d) Sous-échantillonnage par matrice unimodulaire.

d'isoler les singularités rectilignes globalement verticales et globalement horizontales dans deux sous-bandes de taille $2^n \times 2^n$. Pour conserver le même nombre d'échantillons que la sous-bande d'origine, les deux sous-bandes sont ensuite sous-échantillonnées sur deux lattices quinconces complémentaires donnant deux sous-bandes de taille $2^n \times 2^{n-1}$ (ou $2^{n-1} \times 2^n$). Cette opération est réversible car la paire de filtres est duale. De plus, comme illustré sur la figure 2.6 (b), elle a pour effet d'aligner les directions $\pm\pi/4$ le long de l'horizontale ou de la verticale. En appliquant à nouveau la paire de filtres en éventail sur ces deux sous-bandes, on peut alors isoler les singularités orientées autour des directions $\{k\pi/4\}_{k=0...3}$ dans 4 sous-bandes de dimensions $2^n \times 2^{n-1}$. Ces 4 sous-bandes sont à nouveau sous-échantillonnées sur des lattices quinconces pour générer 4 sous-bandes de dimensions $2^{n-1} \times 2^{n-1}$. Ceci a pour effet de replacer les contours à leur position d'origine (figure 2.6 (c)).

Les étapes suivantes de la décomposition s'effectuent de manière légèrement différente. Les 4 sous-bandes sont sous-échantillonnées par quatres matrices dites unimodulaires qui ont pour effet de transformer des images en des parallélogrammes (figure 2.6 (d)). Ceci permet de ré-orienter les contours selon de nouvelles directions. La paire de filtres en éventail est appliquée aux 4 parallélogrammes pour créer 8 sous-bandes isolant des contours orientés autour des directions $\{k\pi/8\}_{k=0...7}$. Ces 8 sous-bandes sont enfin sous-échantillonnées sur une lattice quinconce pour générer 8 sous-bandes de dimensions $2^{n-2} \times 2^{n-2}$. Pour multiplier par deux le nombre de directions d'analyse, ces étapes peuvent être reproduites sur chaque groupe de 4 sous-bandes, et ainsi de suite jusqu'à obtention du nombre de sous-bandes souhaitées (correspondant nécessairement à une puissance de 2). Ce nombre est fonction de l'échelle de la sous-bande d'origine

dans la pyramide Laplacienne. Suivant l'heuristique adoptée pour les Curvelets, les auteurs proposent de multiplier le nombre d'orientations par 2 toutes les 2 échelles en montant vers les hautes fréquences. La partition du domaine fréquentiel obtenue est illustrée figure 2.5 (b).

En termes d'approximation non linéaire, la transformée en Contourlets affiche les mêmes performances que la transformée en Curvelets, à savoir une quasi-optimalité pour les images de classe $\mathcal{C}^2\backslash\mathcal{C}^2$. De plus, l'implémentation proposée par Do et Vetterli permet de borner le facteur de redondance à 1.33. A notre connaissance, les principaux papiers décrivant la transformée en Contourlets présentent des résultats d'approximation non linéaire sur des images tests mais pas de performance en termes de compression. Il est donc difficile d'évaluer les performances des Contourlets par rapport à un schéma de codage par ondelettes classique comme JPEG2000. A partir des résultats d'approximation, on peut déduire que la transformée en Contourlets permet une bonne préservation des contours rectilignes à bas débits mais que la redondance détériore les performances dans les hauts débits.

Il existe bien d'autres représentations non adaptatives d'une image. Par exemple les ondelettes complexes de Kingsbury [Kin98] ou la transformée cortex de Watson [Wat87]. L'étude des Curvelets et des Contourlets met en avant deux limites des bases fixes :

Limite de représentation : il semble très difficile de pouvoir représenter une large classe de singularités géométriques avec un dictionnaire d'atomes fixe sans faire exploser la taille de ce dictionnaire. On peut s'intéresser à des techniques de poursuites de vecteurs [MZ93] pour sélectionner un sous-dictionnaire adapté à une image particulière. Cependant, ces techniques sont reconnues pour être lourdes. En outre, une telle poursuite fait perdre le caractère non adaptatif de la méthode.

Limite de conception : les techniques non adaptatives présentées gagnent la propriété directionnelle mais perdent la propriété de séparabilité de l'ondelette 2D standard. D'autre part, nous répétons que ces transformées sont nées d'un raisonnement dans le domaine fréquentiel pour des signaux continus. Tout ceci se traduit par une plus grande complexité en termes de conception dans le domaine spatial discrétisé.

Les techniques *adaptatives* présentées dans les deux sections suivantes tentent de trouver des solutions à ces limites en s'appuyant sur le noyau d'ondelette séparable. L'idée directrice est de modéliser localement le flux géométrique à l'aide d'un nombre limité de paramètres et de déformer le noyau séparable en conséquence. Pour ce faire, il faut au préalable choisir un *modèle* de géométrie. Dans la section 2.2, nous nous penchons sur les outils permettant de modéliser *localement* la déformation de l'ondelette. Nous verrons en section 2.3 comment ces outils sont utilisés pour parvenir à une modélisation *globale* sur toute l'image. Dans cette section, nous présenterons aussi le maillage 2D comme modèle de ré-échantillonnage d'une image qui peut aussi être vu comme un modèle global de géométrie. Les techniques d'estimation associées à ces modèles (locaux et globaux) sont aussi présentées.

2.2 Modélisations géométriques locales

2.2.1 Directionlets : raisonnement sur lattices

2.2.1.1 Modèle géométrique

La représentation en Directionlets proposée par Velisavljević [Vel05b, VBLVD06] suppose que la géométrie dans un bloc peut être modélisée par des contours rectilignes orientés selon deux directions au maximum. Ces deux directions sont caractérisées par les pentes θ_1 et θ_2 des contours, $\theta_1 \neq \theta_2$. La seule contrainte sur les paramètres θ_1 et θ_2 est qu'ils doivent être rationnels. Le coût de codage de ces paramètres dépend du nombre d'orientations permises. Dans son application au codage [VBLVD06], l'auteur permet 4 orientations $(0, \pi/4, \pi/2, 3\pi/4)$. L'estimation du modèle se fait en observant, pour un multiplicateur de Lagrange donné λ le compromis débit-distorsion obtenu en effectuant la décomposition pour chaque possibilité. Voyons en quoi consiste cette décomposition.

2.2.1.2 Filtrage directionnel sur lattices de $\mathbb{Z}^2$

Effectuer un raisonnement sur les lattices de $\mathbb{Z}^2$ permet de construire un filtrage directionnel n'utilisant que les échantillons de l'image et de conserver la même complexité qu'une décomposition par ondelettes. Supposons que les paramètres choisis peuvent s'écrire comme $\theta_1 = a_1/b_1$ et $\theta_2 = a_2/b_2$ avec $a_1, b_1, a_2, b_2 \in \mathbb{Z}$. Soit Λ une sous-lattice de $\mathbb{Z}^2$ composée des échantillons obtenus par combinaison linéaire des deux vecteurs $\mathbf{d_1} = (a_1, b_1)$ et $\mathbf{d_2} = (a_2, b_2)$. Elle peut être représentée par la matrice $\mathbf{M}_\Lambda$ dite génératrice :

$$\mathbf{M}_\Lambda = \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix} = \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix}, \quad \text{avec} \quad a_1, a_2, b_1, b_2 \in \mathbb{Z}$$

La figure 2.7 montre à gauche un contour de pente $r = 1/2$ discrétisé. Supposons que

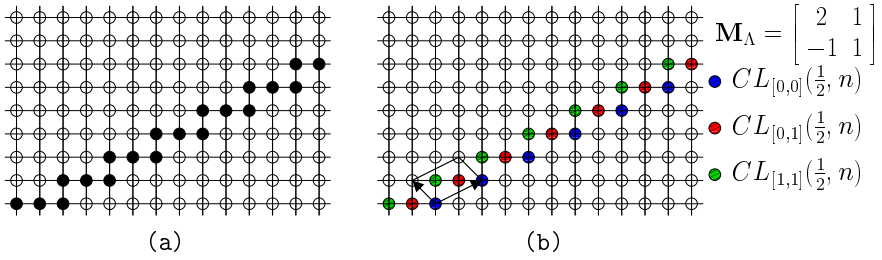


Fig. 2.7 : (a) Un contour discrétisé de pente $r = 1/2$, (b) Co-lignes générées par intersection avec les cosets de la lattice Λ choisie.

l'estimation de géométrie aboutisse à un paramètre $\theta_1 = r$ (cas idéal). Comme un seul contour est présent, le paramètre θ_2 peut être choisi quelconque. La figure 2.7 montre à droite les vecteurs générateurs d'une lattice Λ en choisissant $\theta_1 = r$. Comme expliqué par Velisavljević [VBLVD06], la lattice cubique $\mathbb{Z}^2$ peut être partitionnée en $|\det(\mathbf{M}_\Lambda)|$ classes d'équivalence de la lattice Λ (trois classes dans l'exemple donné). Chaque classe

est déterminée par un vecteur de translation $\mathbf{s}_k$, $k = 0, 1, \cdots, |\det(\mathbf{M}_\Lambda)| - 1$. Une *co-ligne* (« coline » dans le texte original) $CL_{\mathbf{s}_k(\theta_1,n)}$ est définie comme l'intersection entre la classe k et la ligne discrète notée $L(\theta_1, n)$ de pente θ_1 et d'ordonnée à l'origine n. $L(\theta_1, n)$ est totalement représentée par les $|\det(\mathbf{M}_\Lambda)|$ co-lignes $CL_{\mathbf{s}_k(\theta_1,n)}$, $k = 0, 1, \cdots, |\det(\mathbf{M}_\Lambda)| - 1$.

Sur la figure 2.7, nous observons que si le paramètre θ_1 est choisi égal à la pente r du contour, alors chaque co-ligne du contour discrétisé est régulière (tous les échantillons de la co-ligne sont noirs, il n'y a donc pas de discontinuité). L'auteur propose donc d'appliquer une ondelette 1D le long de toutes les co-lignes. Le filtrage est appliqué indépendamment dans chaque coset. La figure 2.8 montre un bloc cette fois composé de *deux* contours rectilignes de pentes $r_1 = -1/2$ et $r_2 = 3/2$. Dans cette figure la lattice Λ est construite en choisissant $\theta_1 = r_1$ et $\theta_2 = r_2$ (cas idéal). La figure montre aussi la position des coefficients non nuls après un filtrage 1D passe-haut sur les co-lignes $CL_{\mathbf{s}_k(\theta_1,n)}$ avec une ondelette de Haar. On observe qu'après filtrage, seuls des résidus orientés de r_2 subsistent. Après sous-échantillonnage, les échantillons restant sont disposés sur une sous-lattice $\Lambda' \subset \Lambda$. La matrice génératrice de Λ' s'écrit :

$$\mathbf{M}_{\Lambda'} = \begin{bmatrix} 2\mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix}$$

On observe qu'après le sous-échantillonnage, les résidus de corrélation sont situés le long des co-lignes $CL_{\mathbf{s}_k(\theta_2,n)}$. Un deuxième filtrage est donc réalisé le long de ces co-lignes. Ce filtrage séparable permet finalement d'isoler les deux contours de départ dans deux sous-bandes distinctes. Notons que si les directions de filtrage sont l'horizontale et la verticale, la décomposition se ramène à la décomposition standard.

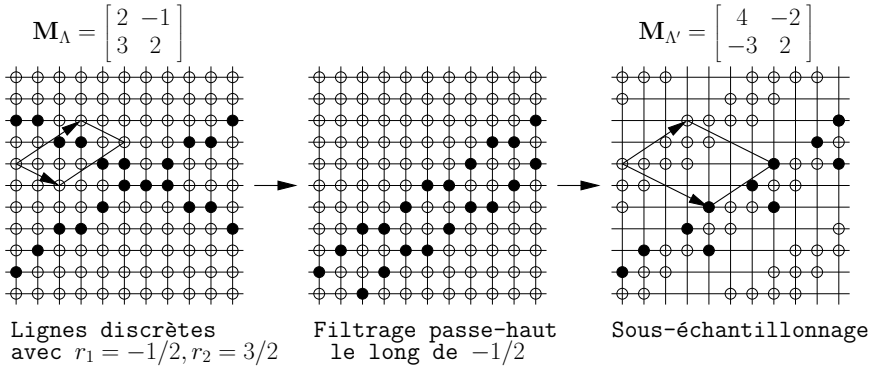


Fig. 2.8 : Filtrage et sous-échantillonnage le long de $-1/2$ dans les cosets de Λ.

2.2.1.3 Modification du ratio d'aspect de l'ondelette

Considérons un bloc constitué de contours rectilignes de pentes r_1 et r_2. Supposons que ce bloc est décomposé comme expliqué précédemment en choisissant les paramètres idéaux $\theta_1 = r_1$ et $\theta_2 = r_2$. Velisavljević et al. [VBLVD06] s'intéressent au nombre de

coefficients non nuls M générés par cette décomposition. Si le bloc est de dimension $N \times N$, l'ordre de grandeur de ce nombre est :

$$M = O((k_1 + k_2)N) \tag{2.1}$$

où k_1 et k_2 sont les nombres de contours rectilignes respectivement de pente r_1 et r_2. On suppose $k_1 \geqslant k_2$. Les auteurs décrivent alors une nouvelle transformée, appelée « Anisotropic Wavelet Transform » qui permet d'améliorer ce résultat en jouant sur les niveaux de décomposition d'ondelettes J_1 et J_2 dans les directions θ_1 et θ_2. Une étape de la transformée se réalise en effectuant J_1 niveaux de décomposition le long des co-lignes orientées de θ_1 et J_2 niveaux le long des co-lignes orientées de θ_2. Les décompositions se font alternativement dans les deux directions. La figure 2.9 montre un exemple de décomposition en prenant $J_1 = 2$ et $J_2 = 1$.

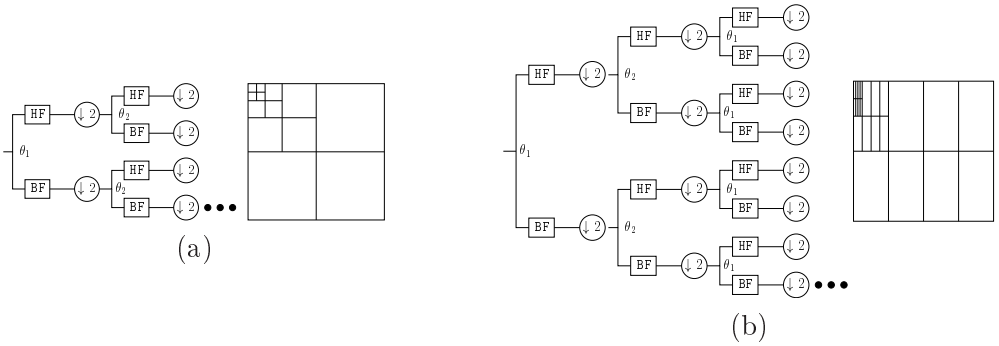


Fig. 2.9 : (a) Ratio de décomposition standard : $J_1 = J_2$, (b) Modification du ratio, ici $J_1 = 2$ et $J_2 = 1$.

Le rapport $\rho = J_1/J_2$ détermine le ratio d'aspect de l'ondelette à chaque échelle. Avec ce ratio, l'auteur montre que le nombre de coefficients non nuls devient :

$$M = O((ak_1 + \frac{1}{a}k_2)(\log_2 N)^2), \quad \text{avec} \quad a = \frac{2^{J_2} - 1}{2^{J_1} - 1} \tag{2.2}$$

Ce résultat peut être vu comme une généralisation de celui obtenu pour la décomposition dyadique classique. On voit ici que le gain par rapport au cas classique est directement lié à la direction d'élongation principale choisie pour l'ondelette : si $k_1 > k_2$, il faut avoir $J_1 > J_2$ c'est à dire allonger le noyau dans la direction θ_1.

2.2.1.4 Approximation non linéaire

Les atomes de représentation obtenus en combinant filtrage directionnel et modification du ratio d'aspect sont appelés Directionlets. Velisavljević et al. [VBLVD06] donnent un résultat d'approximation non linéaire pour la transformée en Directionlets. Ce résultat est estimé pour un *modèle Horizon*, c'est-à-dire une fonction bidimensionnelle I composée de deux parties de régularité $\mathcal{C}^2$ séparées par une discontinuité courbe 1D de

régularité $\mathcal{C}^2$. En segmentant une telle image et en appliquant la transformée en Directionlets sur chaque bloc, les auteurs montrent alors que la meilleure approximation non linéaire de I avec M termes donne une erreur théorique du type :

$$\|I - \tilde{I}_M\|^2 = O(M^{-\alpha}), \quad \text{avec} \quad \alpha \approx 1,562$$

Cette meilleure approximation est obtenue en choisissant un ratio d'aspect $\rho^\star = \alpha$. Ce ratio ne peut être atteint en pratique du fait du caractère discret de la transformée. Mais les auteurs indiquent qu'un ratio $\rho = 3/2$ constitue une bonne approximation de la transformée optimale.

Notons que ce résultat d'approximation n'est valable que si la géométrie de l'image peut être segmentée en contours rectilignes. Dès que les directions de la lattice Λ choisie ne correspondent plus au flux géométrique alors le nombre de coefficients non nuls engendrés est du même ordre que celui obtenu avec la décomposition standard.

2.2.2 Lifting directionnel sur lattice quinconce

Dans la décomposition par ondelettes, une étape de lifting le long de l'axe horizontal s'effectue en séparant les échantillons de l'image en deux composantes polyphases (voir en particulier les travaux de Sweldens, Schröder et Daubechies [SS96, Swe97, DS98] consacrés au schéma lifting). Ces composantes sont les 2 cosets d'une lattice carrée dont une matrice génératrice s'écrit :

$$\mathbf{M} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.3}$$

Pour effectuer une décomposition directionnelle, l'idée de Chappelier et al. [CGM04b] est de définir les 2 composantes polyphases sur les deux cosets d'une lattice quinconce. Une telle lattice est donnée par la matrice génératrice :

$$\mathbf{M} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

Les deux cosets seront dits pair et impair et leurs échantillons seront notés I_e et I_o. Un niveau de décomposition 1D s'opère comme suit. Comme dans le schéma standard, la sous-bande haute fréquence est calculée sur le coset impair en prédisant un échantillon avec ses voisins dans le coset pair. Cependant, comme nous le voyons sur la figure 2.10, un échantillon impair est entouré par quatre voisins pairs. C'est ici que les auteurs intègrent une dose d'adaptivité à la transformée : ils autorisent, via un paramètre θ binaire, la prédiction d'un échantillon impair soit à l'aide de ses voisins horizontaux, soit à l'aide de ses voisins verticaux. La basse fréquence est calculée sur le coset pair en mettant à jour un échantillon à l'aide des détails voisins précédemment calculés dans le coset impair. Dans le schéma standard, un échantillon impair est utilisé exactement deux fois pour prédire des échantillons pairs. Ici, chaque échantillon impair peut être utilisé de zéro à quatre fois. Les auteurs proposent donc de modifier les coefficients de mise à jour du lifting en fonction du nombre de fois où l'échantillon impair a été utilisé lors de l'étape de prédiction.

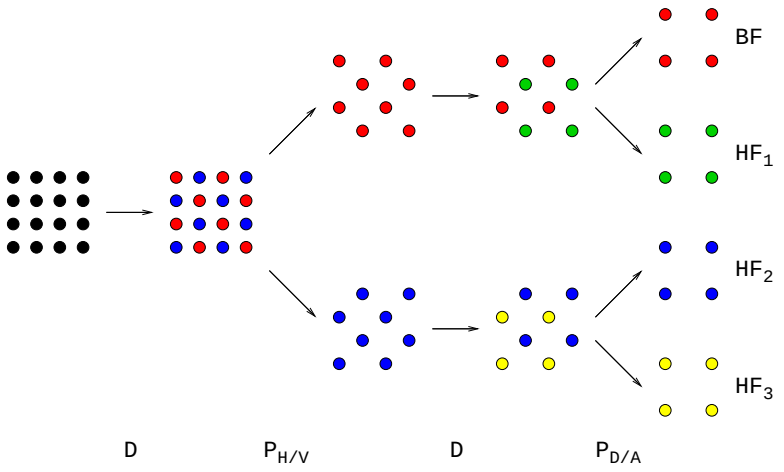


Fig. 2.10 : Décomposition par ondelettes orientées [Cha05b]. D : décomposition polyphase. $P_{H/V}$: Prédiction Horizontale/Verticale. $P_{D/A}$: Prédiction Diagonale/Antidiagonale.

A l'issue de cette décomposition 1D, les détails et les approximations sont situés sur deux grilles quinconces. Chacune de ces grilles quinconces est à son tour séparée en deux composantes polyphases pour poursuivre la décomposition. Cette fois-ci, les échantillons pairs et impairs sont situés sur des grilles carrées mais décalées d'un angle de $\pi/4$. Les étapes de prédiction et de mise à jour sont réalisées comme précédemment mais chaque échantillon impair peut désormais être prédit à l'aide de ses voisins diagonaux situés le long de la direction diagonale ou le long de la direction antidiagonale. A la fin de cette décomposition, l'image de départ est représentée en une sous-bande basse fréquence et trois sous-bandes hautes fréquences. La représentation est non redondante et réversible. Notons cependant que si la décomposition permet d'orienter les directions de filtrage, elle ne propose pas d'outils pour modifier le ratio d'aspect de l'ondelette.

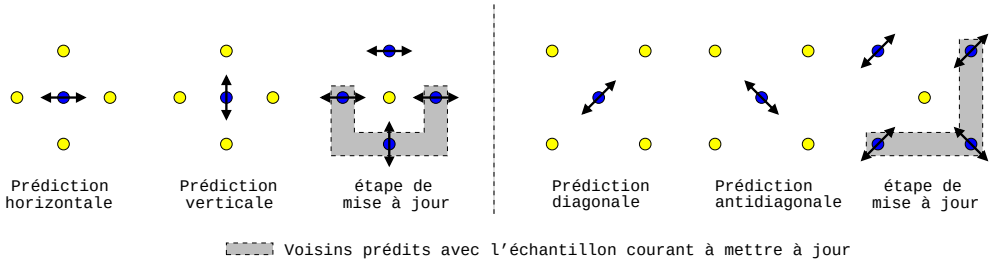


Fig. 2.11 : Etapes de prédiction et de mise à jour au niveau quinconce et au niveau carré.

Le modèle géométrique utilisé dans ce schéma est constitué de deux cartes d'orientation représentant les décisions binaires θ pour la décomposition 1D. Ces cartes sont calculées pour le premier niveau et sont ré-utilisées pour les niveaux suivants. Transmettre une décision pour chaque échantillon impair serait trop coûteux. Les auteurs

décident donc de regrouper les échantillons en blocs de taille minimale 16×16. Une structure en Quadtree (voir section 2.3) permet en outre d'agréger des blocs voisins lorsque le contenu local est régulier. Dans un bloc donné, chaque décision est testée.

Dans [PPPP05], Piella et al. introduisent une technique adaptative qui s'appuie également sur une décomposition quinconce mais les étapes de prédiction et de mise à jour s'effectuent différemment. Un échantillon pair est prédit en faisant la moyenne de ses quatre voisins. L'adaptivité intervient lors de l'étape de mise à jour. Un échantillon impair peut être soit mis à jour avec ses voisins pairs dans une fenêtre 5×5 (figure 2.12) soit rester inchangé. Une décision est prise en calculant une *semi-norme*[1] du gradient dans la fenêtre locale puis en comparant cette semi-norme à un seuil. Si la valeur est supérieure au seuil alors l'échantillon reste inchangé. Si la valeur est inférieure au seuil, alors la région est considérée comme régulière et l'échantillon est mis à jour. Les coefficients de mise à jour sont calculés de sorte que la décision prise à l'analyse puisse être retrouvée à la synthèse. L'approche est donc en marge par rapport aux précédentes car elle ne nécessite pas de transmettre un modèle géométrique. Notons que le fait de modifier l'étape de mise à jour peut être vu comme une manière d'adapter le ratio d'aspect du noyau de représentation. Un filtre long cause de nombreux rebonds autour des singularités lors d'une approximation. En réduisant la taille du filtre dans les régions à fort gradient, on évite ce phénomène. Notons cependant que le filtrage proposé n'est pas directionnel.

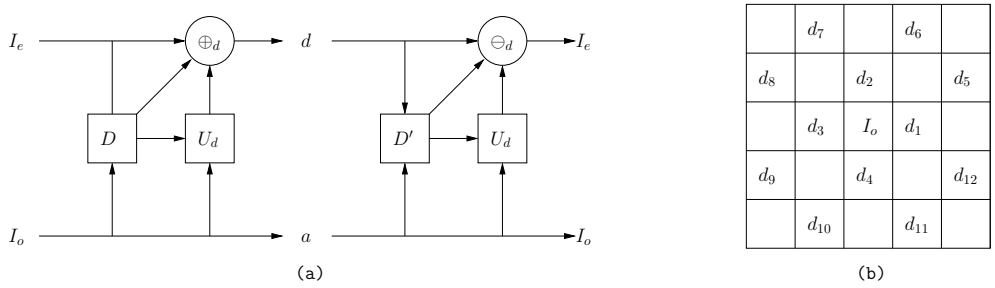


Fig. 2.12 : Schéma lifting adaptatif [PPPP05]. (a) La mise à jour dépend d'une décision D, (b) Exemple de voisinage utilisé lors de l'étape de mise à jour.

2.2.3 Lifting directionnel pour un filtrage sous-pixellique

Les outils précédents permettent d'effectuer une décomposition directionnelle de même complexité que la décomposition ondelettes en s'appuyant uniquement sur les échantillons d'origine. Pour mieux capturer les contours réels, il est cependant intéressant de s'appuyer sur un échantillonnage plus fin que la grille des pixels. La séparation en composantes polyphases dans le schéma lifting permet de le faire tout en conservant la propriété de reconstruction parfaite.

[1] Si p est une semi-norme et x un signal, alors $p(x) = 0$ n'implique pas $x = 0$. Ceci distingue p d'une norme.

Dans [DWL04, WZVS06, DWW+07], les auteurs proposent ainsi un schéma de lifting directionnel autorisant des prédictions et mises à jour sous-pixelliques. Dans un bloc b de l'image, la géométrie est modélisée par des contours rectilignes pouvant prendre deux orientations : une orientation globalement horizontale θ_h et une orientation globalement verticale θ_v (voir figure 2.13). Le coût de codage d'une orientation dépend du nombre d'orientations permises. Le schéma de Wang et al. [WZVS06] autorise un choix parmi 5 orientations correspondant à une précision au demi pixel. Celui de Ding et al. [DWL04, DWW+07] autorise une précision arbitraire mais en pratique les auteurs se limitent au quart de pixel.

Dans les méthodes citées précédemment, remarquons que l'ordre de la décomposition (filtrage globalement horizontal suivi d'un filtrage globalement vertical, ou l'inverse) est choisi arbitrairement et les deux paramètres θ_h et θ_v sont estimés dans le cadre d'une optimisation lagrangienne en testant toutes les possibilités. Dans [JRB07a, JRB07c], Jeannic et al. précisent que ce choix arbitraire peut conduire à une décorrélation sous-optimale du bloc, en particulier si la direction de régularité maximale réelle ne fait pas partie des orientations candidates pour le premier filtrage. Ils proposent ainsi de choisir l'ordre de décomposition à l'intérieur d'un bloc de façon adaptative en s'appuyant sur une métrique locale (le gradient) pour détecter les contours. Dans ce cas, la première orientation de filtrage dans chaque bloc correspond toujours à la direction de régularité maximale. Au lieu de transmettre les paramètres θ_h et θ_v pour chaque bloc, les auteurs proposent en outre d'encoder les contours détectés [JRB07b]. Ceci leur permet de reproduire les décisions au décodage.

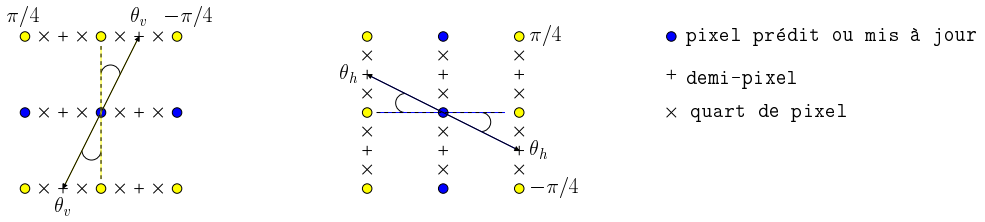


Fig. 2.13 : Une étape de lifting (prédiction ou mise à jour) dans une direction θ_v et θ_h. Les orientations possibles correspondent à une précision au pixel, demi-pixel ou quart de pixel. En pratique θ_v et θ_h sont limités à l'intervalle $[-\pi/4, \pi/4]$ autour de la verticale et de l'horizontale.

Etant données deux orientations θ_h et θ_v estimées à l'intérieur d'un bloc, supposons à présent qu'une décomposition globalement verticale soit tout d'abord réalisée en séparant les échantillons en deux composantes polyphases comme dans le cas classique. Nous rappelons que ceci revient à définir les deux signaux I_e et I_o par :

$$\begin{cases} I_e(x,y) = I(x,2y) \\ I_o(x,y) = I(x,2y+1) \end{cases} \tag{2.4}$$

Dans la décomposition standard, les échantillons impairs sont toujours prédits l'aide de leurs deux voisins pairs verticaux. L'idée est d'effectuer la prédiction dans la direction

θ_v en interpolant les échantillons pairs. La haute fréquence calculée sur un échantillon impair s'écrit alors :

$$I_o(x,y) = I_o(x,y) + \alpha_0 I_e(x - \tan\theta_v, y+1) + \alpha_1 I_e(x + \tan\theta_v, y) \tag{2.5}$$

où α_0 et α_1 sont les coefficients de l'étape de prédiction courante. La prédiction verticale en lifting effectuée dans le cas standard est un cas particulier de (2.5) en prenant $\theta_v = 0$.

A l'étape de mise à jour, les échantillons pairs sont remplacés par les basses fréquences :

$$I_e(x,y) = I_e(x,y) + \beta_0 I_o(x - \tan\theta_v, y+1) + \beta_1 I_o(x + \tan\theta_v, y) \tag{2.6}$$

où β_0 et β_1 sont les coefficients de mise à jour pour l'étape de lifting courante. Dans [DWW+07], le schéma présenté est très général et n'oblige pas à utiliser le même angle θ_v pour la mise à jour que pour la prédiction. C'est cependant ce que les auteurs font en pratique pour limiter le coût de la géométrie.

Après avoir effectué un filtrage dans la direction θ_v, un bloc est décomposé en deux sous-bandes. Ces deux sous-bandes sont séparées en composantes polyphases, les colonnes paires et impaires, et subissent une décomposition orientée de θ_h de façon similaire à la décomposition verticale. A l'issue des filtrages globalement vertical puis globalement horizontal, un bloc est décomposé en quatre sous-bandes comme dans le cas classique. Si un bloc contient des contours rectilignes réellement orientés le long de θ_h et θ_v alors ces contours se retrouvent isolés dans une et une seule sous-bande.

Malgré les interpolations, notons que toutes les étapes de prédictions et de mises à jour sont parfaitement réversibles. En effet, chaque interpolation est effectuée en utilisant exclusivement les échantillons pairs pour une prédiction des échantillons impairs et les échantillons impairs pour une mise à jour des échantillons pairs. Puisque les orientations θ_v et θ_h prennent leur valeur dans un ensemble fini connu a priori, toutes les positions possibles des échantillons à interpoler sont connues. Alors, le choix d'un filtre interpolateur permet de déterminer à l'avance tous les paramètres d'interpolation nécessaires. Ceci permet de limiter l'incrément de complexité lié au ré-échantillonnage. Dans [DWW+07], les auteurs font le choix d'une interpolation Sinc [Yar02]. Notons que le choix de l'interpolateur peut jouer sur les performances et la qualité du filtrage directionnel effectivement appliqué.

2.2.4 Bandelettes pour un suivi des lignes de flux

Dans les technologies précédentes, l'exploitation des régularités se fait au niveau du pixel en considérant chaque vecteur de flux indépendamment. Il n'y a donc pas réellement de *suivi* d'une ligne de flux par intégration des vecteurs. Les Bandelettes, introduites par Le Pennec et Mallat [PM00, Pen02, PM05] permettent un tel suivi même le long de contours *courbes*.

2.2.4.1 Flux géométrique parallèle

Dans un bloc donné de l'image, la construction des Bandelettes s'appuie sur un modèle de flux constant dans une des deux dimensions (figure 2.14). Un flux constant le long de y est dit *parallèle verticalement*. Un flux constant le long de x est dit *parallèle horizontalement*. Chaque flux permet d'intégrer un réseau de courbes de flux parallèles soit à une fonction de x, notée $c(x)$, ou soit à une fonction de y , notée $c(y)$.

Supposons que l'on souhaite modéliser le flux par des vecteurs parallèles verticalement. Comme ces vecteurs ne dépendent que de x, ils peuvent être écrits : $\gamma(x, y) = \gamma(x) = (1, c'(x))$ où c' est la dérivée de la courbe de flux $c(x)$.

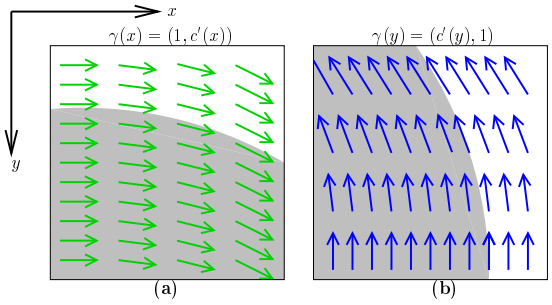


Fig. 2.14 : (a) Flux parallèle verticalement, (b) Flux parallèle horizontalement.

Pour pouvoir représenter des contours courbes réguliers de formes polynômiales variées, les auteurs proposent alors de modéliser la dérivée $c'(x)$ comme une somme de B-splines d'ordre m $B_m(x)$ translatées[2] et dilatées d'un facteur d'échelle 2^l :

$$c'(x) = \sum_{p=1}^{P} \theta_p B_m(2^{-l}x - p) \tag{2.7}$$

Les coefficients θ_p et l'exposant l sont les paramètres du modèle à optimiser. Le facteur d'échelle 2^l définit la régularité du flux et le nombre P de coefficients. Si le bloc b a une largeur de 2^k, alors on a $1 \leqslant 2^l \leqslant 2^k$ et $k+1$ valeurs peuvent être testées pour l'exposant l. Pour un exposant l donné, le nombre P de paramètres θ_p pour ce bloc est 2^{k-l}. La figure 2.15 montre des fonctions $c'(x)$ obtenues en sommant P B-splines d'ordre 1 translatées avec des coefficients θ_p arbitraires. La courbe de flux $c(x)$ intégrée à partir de ce flux est aussi représentée. On voit qu'en augmentant le nombre de paramètres, il est possible de représenter des courbes de flux polynômiales de formes variées.

Dans [PM05], les paramètres θ_p sont calculés en minimisant une énergie de flux dans chaque bloc b. Intuitivement, chaque vecteur du flux optimal doit être orienté dans la direction où l'image est la plus régulière. Cette intuition est traduite mathématiquement par les auteurs comme la minimisation de :

[2] Une B-spline d'ordre m est obtenue en convoluant m fois la fonction indicatrice $\mathbf{1}_{[-1/2,1/2]}$ avec elle-même.

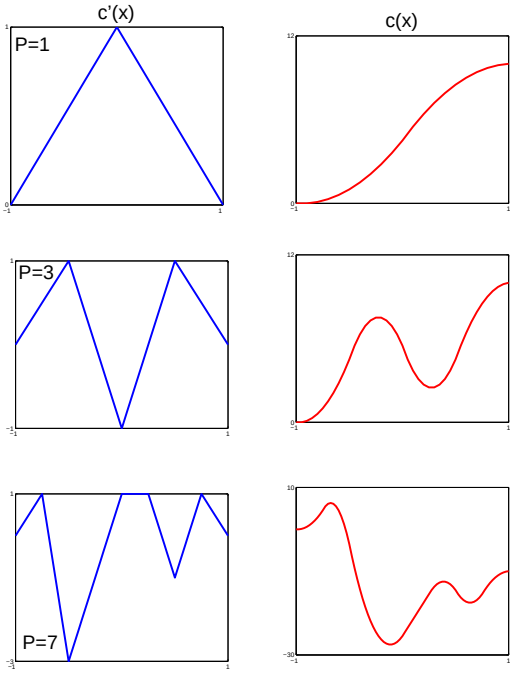


Fig. 2.15 : A gauche, courbes $c'(x)$ obtenues en sommant P B-spline linéaires translatées. A droite, les courbes de flux obtenues en intégrant $c'(x)$.

$$\mathbf{E}(\Gamma) = \int_b \left| \frac{\partial (I * \phi)(x,y)}{\partial \gamma(x,y)} \right|^2 \mathrm{d}x\mathrm{d}y \tag{2.8}$$

où ϕ est un filtre lissant. Si le flux géométrique est choisi parallèle verticalement alors l'expression se simplifie en :

$$\mathbf{E}(\Gamma) = \int_b \left| I * \frac{\partial \phi}{\partial x}(x,y) + c'(x) I * \frac{\partial \phi}{\partial y}(x,y) \right|^2 \mathrm{d}x\mathrm{d}y \tag{2.9}$$

On voit que cette énergie de flux ne dépend que des paramètres θ_p qui caractérisent la dérivée $c'(x)$. Les paramètres optimaux (au sens de cette énergie) sont calculés en résolvant le système linéaire qui s'obtient en annulant la dérivée de $\mathbf{E}(\Gamma)$. Un raisonnement similaire permet de déterminer les paramètres optimaux d'un flux parallèle horizontalement.

Pour un bloc donné dans l'image, il n'est pas possible de savoir à l'avance si le flux réel est mieux modélisé par un flux parallèle horizontalement, parallèle verticalement, ou encore s'il est plus avantageux de ne coder aucun flux. Chaque configuration est testée. Celle aboutissant au meilleur compromis débit-distorsion pour un multiplicateur lagrangien donné est retenue. Dans leur implémentation, les auteurs utilisent la B-spline d'ordre 1. Le pas de quantification des coefficients est choisi de sorte à autoriser une précision du champ de vecteurs de l'ordre du 1/8 de pixel. Nous voyons au paragraphe suivant comment les auteurs exploitent un tel champ en pratique.

2.2.4.2 Rectification des contours par déformation du bloc

Nous supposerons dans ce paragraphe et le suivant que la géométrie dans le bloc b considéré est modélisée par un flux parallèle verticalement. Etant donné un tel flux, les auteurs cherchent à effectuer un filtrage exactement le long des courbes de flux. Comme la courbe $c(x)$ prend des valeurs dans $\mathbb{R}$, un ré-échantillonnage préalable du bloc est nécessaire. Le Pennec [Pen02] définit ainsi la transformation w suivante

$$\begin{aligned} w : \mathcal{D} &\to \tilde{\mathcal{D}} \\ (x, y) &\mapsto (x, y - c(x)) \end{aligned} \tag{2.10}$$

où $\mathcal{D}$ est l'ensemble des pixels du bloc et $\tilde{\mathcal{D}}$ un domaine déformé inclus dans $\mathbb{R}^2$. Cette déformation aligne le long de l'axe horizontal une courbe $\mathcal{C}$ définie par l'ensemble des points $\{(x, c(x) + K\}$ où K est une constante. Elle préserve les directions verticales. A partir de w, l'auteur définit ensuite un opérateur de déformation $\mathbf{W}$ qui agit sur l'image de sorte que ses singularités soient alignées sur l'axe horizontal :

$$\mathbf{W}I(x, y) = I(w^{-1}(x, y)) = I(x, y + c(x)) \tag{2.11}$$

En pratique l'opérateur $\mathbf{W}$ revient à translater chaque colonne x du facteur de translation $c(x)$ le long de l'axe vertical. Comme on le voit sur la figure 2.16, cette translation aligne le flux et le contour le long de l'axe horizontal. *Tout se passe comme si le contour était rectifié pour s'adapter à une décomposition horizontale/verticale*. Comme chaque échantillon dans une ligne du bloc déformé correspond à une courbe de flux dans le domaine d'origine $\mathcal{D}$, un filtrage le long de l'axe horizontal dans $\tilde{\mathcal{D}}$ revient à filtrer le long de la courbe dans $\mathcal{D}$. Un filtrage vertical permet ensuite de décomposer le bloc déformé en quatre sous-bandes comme dans la décomposition classique. Comme w préserve les directions verticales, ceci revient également à filtrer le long de l'axe vertical dans le domaine d'origine.

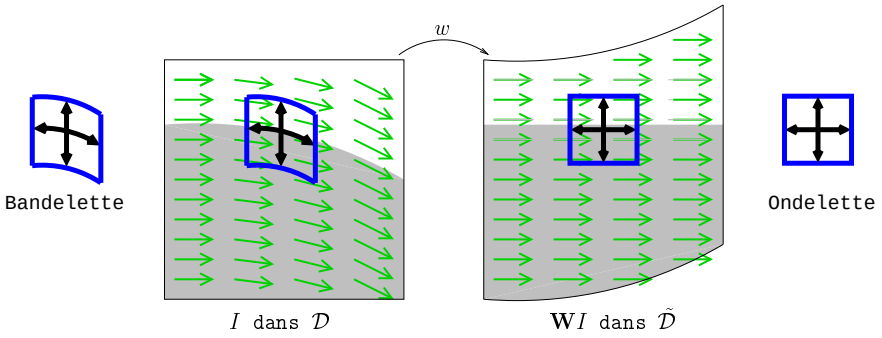


Fig. 2.16 : Rectification du flux géométrique pour une décomposition horizontale/verticale.

Dans sa thèse, Le Pennec [Pen02] prouve que, dans le cas *continu*, projeter $\mathbf{W}I$ sur un atome séparable $\psi(x) \otimes \psi(y)$ est équivalent à projeter I sur un atome déformé

$\psi(x) \otimes \psi(y - c(x))$ qu'il appelle une *Bandelette*. Cette bandelette permet un filtrage séparable le long de l'axe vertical et le long de la courbe de flux. Tout le raisonnement reste bien sûr valable dans le cas d'un flux parallèle horizontalement. Dans le cas discret, cette équivalence n'est pas juste car la déformation de l'image nécessite un ré-échantillonnage irréversible : les valeurs des nouveaux échantillons sont calculées en interpolant les valeurs des pixels d'origine qui elles sont perdues définitivement. Seul un filtre interpolateur de type sinus cardinal à support infini rendrait cette opération réversible, mais un tel filtre ne peut pas être implémenté en pratique. Il faut préciser que ces pertes sont des pertes *numériques* : la question qui importe est de savoir si ces pertes sont visibles à l'œil. Pour mener à bien les translations, Le Pennec préconise la méthode d'interpolation à base de splines proposée dans [BTU01]. Avec cette méthode, il qualifie la déformation de l'image de « quasi réversible ».

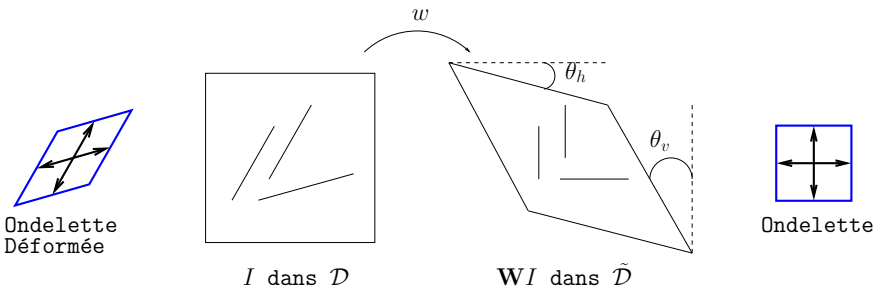


Fig. 2.17 : Rectification à deux paramètres de Taubman et Zakhor [TZ94b].

L'idée de rectifier la géométrie d'une image pour permettre un filtrage horizontal/vertical avait déjà été développée auparavant par Taubman et Zakhor [TZ94b]. Dans leurs travaux, la géométrie dans un bloc est modélisée par des contours rectilignes orientés selon deux angles θ_h et θ_v comme dans les approches de lifting directionnel vues plus haut. Ce modèle à deux paramètres permet de définir une transformation w qui déforme un bloc en un parallélogramme autour de son centre de gravité comme illustré figure 2.17. Dans ce cas l'opérateur $\mathbf{W}$ aligne des contours rectilignes orientés de θ_h et θ_v le long de l'axe horizontal et le long de l'axe vertical en translatant chaque ligne y d'un facteur $y \tan \theta_v$ et chaque colonne x d'un facteur $x \tan \theta_h$. Le choix du couple (θ_h, θ_v) se fait en sélectionnant des couples candidats par détection de contours puis en testant tous les couples candidats.

2.2.4.3 Elongation de l'ondelette : la Bandelettisation

Supposons à nouveau que la géométrie dans un bloc puisse être modélisée par un flux parallèle verticalement. Nous avons vu que la déformation du bloc permet d'aligner les courbes régulières le long de l'axe horizontal. Après projection du bloc déformé sur la base d'ondelettes $\left\{\phi_{J,\mathbf{m}}, \psi^H_{j,\mathbf{m}}, \psi^V_{j,\mathbf{m}}, \psi^D_{j,\mathbf{m}}\right\}_{\mathbf{m},j=1\ldots J}$, l'énergie des contours horizontaux se trouve isolée à chaque échelle dans la sous-bande que nous avons notée V

(figure 2.18(b)). Dans une telle sous-bande, le nombre de coefficients non nuls est directement proportionnel à la dimension des contours. Comme expliqué par Le Pennec et Mallat [PM05], ceci s'explique par le fait que l'ondelette $\psi^V_{j,\mathbf{m}}$ n'a pas de moment nul le long de l'axe x et donc ne tire pas avantage des corrélations horizontales. Pour y remédier, les auteurs proposent d'effectuer une décomposition ondelette 1D le long de chaque ligne de chaque sous-bande V. Ceci permet de modifier les ondelettes $\psi^V_{j,\mathbf{m}}$ pour leur apporter les moments nuls nécessaires le long de l'axe horizontal. Comme on le voit sur la figure 2.18(d), la décomposition 1D permet de compacter l'énergie de la sous-bande sur quelques coefficients. Ce procédé est appelé *Bandelettisation*. Les nouveaux atomes sont allongés d'avantage dans la direction du flux que dans la direction verticale. Leur ratio d'aspect dépend du niveau de décomposition choisi le long de l'axe horizontal. Dans le domaine non déformé ceci revient à construire des atomes dont le support est allongé le long des lignes de flux. Le même raisonnement peut être suivi si la géométrie est modélisée par un flux parallèle horizontalement. Dans ce cas, la bandelettisation doit être opérée sur les noyaux $\psi^H_{j,\mathbf{m}}$.

Dans [PM03] les auteurs montrent que la transformée en Bandelettes fournit une décroissance optimale de l'erreur d'approximation pour les images de type $\mathcal{C}^\alpha \backslash \mathcal{C}^\alpha$, à savoir :

$$\|I - \tilde{I}_M\|^2 \leqslant K \cdot M^{-\alpha} \tag{2.12}$$

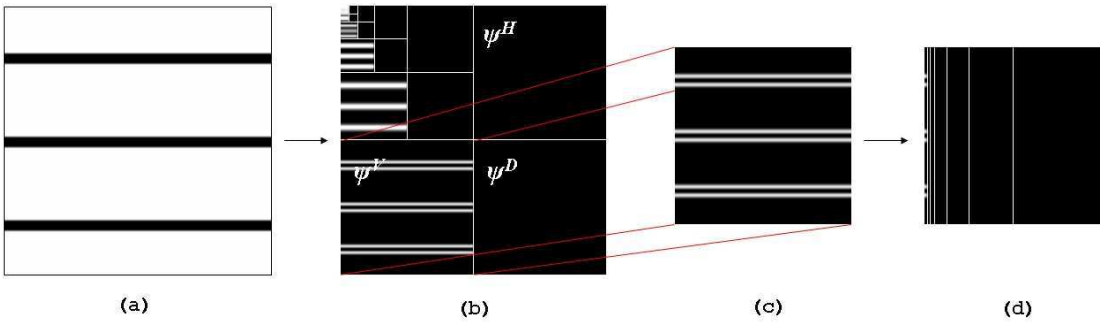


Fig. 2.18 : (a) Image simple de contours horizontaux, (b) Décomposition horizontale/verticale par ondelettes, (c) Sous-bande V, (d) Décomposition 1D le long de l'axe horizontal.

2.2.4.4 Bandelettes seconde génération

Dans sa thèse, Peyré [Pey05b] poursuit les travaux de Le Pennec dans le but de construire une base de bandelettes discrètes adaptée à la grille d'échantillonnage d'origine de l'image. Les *Bandelettes de seconde génération* s'appuient sur une modélisation de la géométrie résiduelle *dans le domaine ondelettes*. Après décomposition préalable de l'image dans une base d'ondelettes standard, chaque sous-bande d'orientation θ et échelle j est partitionnée en blocs. La géométrie résiduelle dans chaque bloc est ensuite modélisée par un flux parallèle comme dans l'approche précédente. La difficulté est de tirer partie de cette géométrie sans recourir à un ré-échantillonnage du bloc. Pour ce

faire, Peyré utilise la transformée de Alpert [Alp92] discrète qui revient à décomposer un bloc en fines bandes dyadiques qui suivent au mieux la géométrie Γ. La construction de la base de Alpert dans le cas général utilise des outils mathématiques évolués dont Peyré donne une interprétation dans un contexte simplifié où les lignes de flux modélisées sont des droites parallèles à une droite notée d et où la transformée de Alpert est construite à partir de l'ondelette de Haar. En considérant un bloc ayant N échantillons au total, cette transformée de Alpert dite d'ordre 0 s'opère en deux temps :

Premier temps : Chaque point de la grille d'échantillonnage du bloc est projeté sur une même droite $d^{\perp}$ orthogonale à d (figure 2.19 d'après Peyré [Pey05b]). Tous les points sont ensuite ordonnés de 0 à $N-1$ en fonction de leur abscisse sur cette droite. Enfin, une fonction 1D est créée : elle est définie sur l'ensemble discret $\{0,\ldots,N-1\}$ et sa valeur en k correspond à la valeur du $k^{ème}$ échantillon dans l'ordre défini précédemment.

Second temps : Si la direction d suit bien un contour traversant le bloc, alors l'étape précédente permet de transformer une discontinuité de type ligne en une discontinuité de type point, de la même façon que la projection de Radon vue au paragraphe 2.1.1. Pour compléter la transformée de Alpert d'ordre 0, il suffit alors de projeter cette fonction 1D sur une base d'ondelettes de Haar. On sait qu'une telle ondelette 1D est efficace pour représenter ce type de signaux.

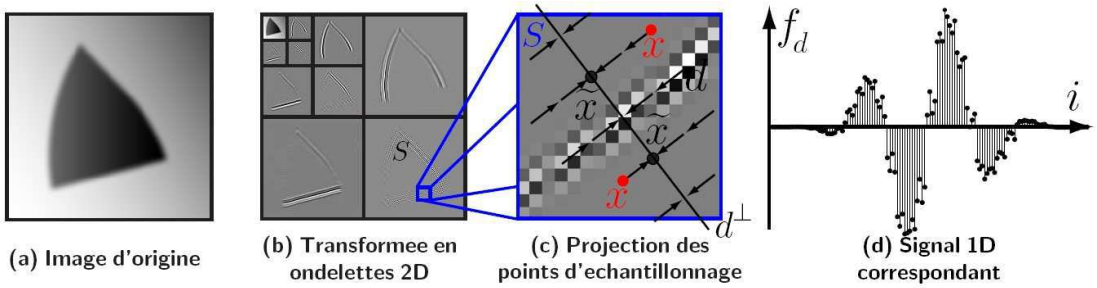


Fig. 2.19 : Réordonnancement discret des points d'échantillonnage. D'après [Pey05b].

Dans un cadre de compression d'images naturelles, Peyré contraint la partition de chaque sous-bande à des blocs de dimensions fixes 4×4. En effet, l'auteur considère qu'on ne peut pas exploiter une régularité géométrique sur une longueur de plus d'une dizaine de pixels à l'échelle de l'image. Et ceci se traduit par des corrélations sur environ 4 pixels à l'échelle $j = 1$. Pour mener à bien la transformée de Alpert sur des blocs de taille 4×4, l'auteur pré-définit 12 ordonnancements possibles des 16 échantillons correspondant aux 12 directions de régularité qu'il choisit de tester. Ces groupements découlent de l'étape 1 de la transformée de Alpert.

2.2.5 Wedgelets : imagettes de contours

La théorie des Wedgelets a été proposée par Donoho [Don99]. Romberg et al. [RWB02] puis Wakin et al. [WRCB02] se sont penchés sur leur application à la compression d'images naturelles. Une Wedgelet ψ est une fonction élémentaire définie de façon adaptative sur un bloc b et comprenant deux régions constantes séparées par une discontinuité rectiligne. La discontinuité sépare le bloc en deux régions $\mathcal{R}_a$ et $\mathcal{R}_b$. Une Wedgelet est caractérisée par 4 paramètres : les 2 points d'intersection (v_1, v_2) de la discontinuité avec les bords du bloc et les 2 valeurs c_a, c_b prises de part et d'autre de cette discontinuité (figure 2.20). Les paramètres c_a et c_b sont déterminés en calculant la moyenne de l'image sur ces deux régions. Notons qu'une Wedgelet constitue à elle seule une approximation d'un bloc. Aucun filtrage n'est ici réalisé.

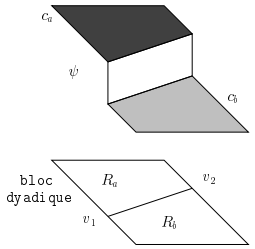


Fig. 2.20 : Une Wedgelet.

Soient b un bloc de l'image et $\mathcal{V}$ une collection de couples (v_1, v_2) choisie au préalable. La décomposition en Wedgelets de $I(b)$ consiste à calculer la Wedgelet correspondant à chaque couple dans $\mathcal{V}$. Lors d'une approximation, la Wedgelet donnant la plus petite erreur sur le bloc est conservée. Une décomposition multi-échelles d'une image peut être obtenue en décomposant $I(b)$ sur tous les blocs dyadiques à tous les niveaux d'une segmentation en Quadtree (les définitions du Quadtree et d'un bloc dyadique seront données dans la section suivante). Pour une image de dimensions $N \times N$, calculer une décomposition multi-échelles complète a une complexité en $O(MN^2 \log_2 N^2)$, où M est la taille de $\mathcal{V}$. En restreignant intelligemment l'ensemble $\mathcal{V}$, il est possible d'utiliser les projections à un niveau fin du Quadtree pour calculer les projections au niveau plus grossier. Ceci permet de ramener la complexité à $O(MN^2)$. Une fois calculées les projections de I sur chaque carré dyadique, il est possible d'utiliser ces projections pour construire une approximation de I en élaguant les branches du quatree. Ceci se fait au cours d'une optimisation débit-distorsion qui sera développée au paragraphe 2.3.2.

En termes d'approximation non linéaire et de compression, les Wedgelets affichent des performances quasi-optimales pour des fonctions composées de régions constantes séparées par des singularités régulières de type $\mathcal{C}^2$. La représentation en Wedgelets est donc particulièrement bien adaptée aux images de type « cartoon ». Ces performances se dégradent lorsqu'il s'agit de représenter des zones texturées. Pour cette raison, Wakin et al. [WRCB02] ont proposé un modèle hybride où un bloc du Quadtree peut être représenté soit avec une Wedgelet soit avec une base d'ondelettes lorsque l'approxima-

tion en Wedgelet échoue. La difficulté est de distinguer les zones texturées des régions homogènes ou de contours. En effet, lorsqu'on examine le résidu entre l'image originale et la meilleure approximation par Wedgelets, on s'aperçoit que ce résidu présente une forte énergie à la fois dans les zones texturées et dans les zones contenant un simple contour. En effet, du fait de la discrétisation des orientations $\mathcal{V}$, un contour est souvent reconstruit avec une petite erreur de localisation. Visuellement, les auteurs notent que cette erreur n'est pas visible ce qui permet d'obtenir un bon résultat subjectif. Mais cette erreur a bien sûr un impact fort sur le PSNR. Dans les travaux que nous avons menés et qui seront présentés au chapitre 4, une difficulté similaire est apparue. En effet, nos travaux s'appuient sur un ré-échantillonnage de l'image d'origine qui provoque des pertes *numériques*. A la reconstruction, ces pertes n'ont pas d'impact sur la qualité visuelle des contours mais elles limitent cependant les valeurs de PSNR.

2.3 Modélisations géométriques globales

Dans la section précédente, nous nous sommes placés au niveau d'un bloc de l'image et avons présenté des outils pour modéliser la géométrie à l'intérieur de ce bloc. Il reste à préciser comment les blocs sont choisis en pratique pour apporter un compromis adaptivité/parcimonie global sur l'ensemble du domaine image. Nous nous y penchons dans la première partie de cette section. Dans la seconde partie, nous nous concentrons sur une modélisation globale de la géométrie d'une image en marge du modèle par blocs : le maillage 2D.

2.3.1 Segmentation du domaine image

La géométrie d'une image est une donnée complexe à modéliser globalement. Pour cette raison, la majorité des méthodes citées à la section précédente raisonnent à un niveau plus local en segmentant le domaine image. Dans ce cas, le modèle de géométrie comprend deux types de paramètres :

- Une partition $\mathcal{B}$ du domaine image en blocs
- Pour chaque bloc, un modèle géométrique local représenté par un ensemble de paramètres $\Theta = \{\theta_p\}_{0 \leqslant p < P}$, avec P fixé. Par exemple, ce modèle peut être l'un de ceux présentés à la section précédente.

Pour créer la partition $\mathcal{B}$, la méthode la plus simple est de segmenter l'image en blocs de taille fixe. Dans ce cas, la partition a un coût nul en termes de coût de codage et le coût de la géométrie repose uniquement sur les ensembles de paramètres Θ calculés pour chaque bloc. Néanmoins, cette partition ne tient pas compte des disparités géométriques pouvant exister à l'intérieur d'une même image. Un bloc trop gros par rapport au contenu géométrique ne permet pas de modéliser correctement ce contenu avec P paramètres. Un bloc trop petit ne permet pas de tirer pleinement partie des régularités. Une solution à ce problème est de segmenter le domaine image avec des blocs de taille maximale (16×16 par exemple) puis d'autoriser un partitionnement plus fin de chaque bloc en fonction de son contenu. Dans [DWL04], Ding et al. proposent ainsi d'utiliser 3

modes de partition (figure 2.21(a)). Le choix du mode est un paramètre supplémentaire du modèle.

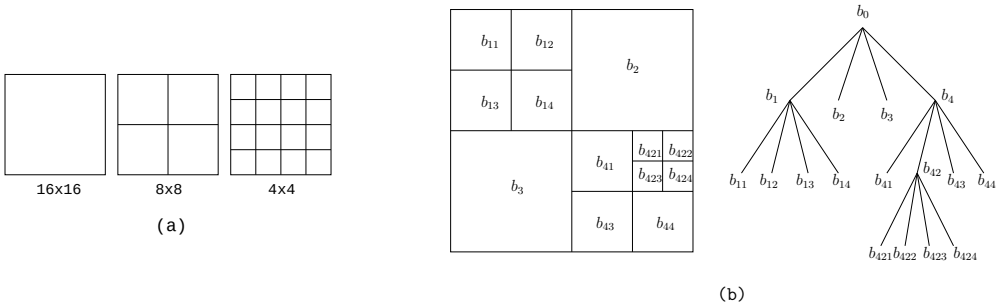


Fig. 2.21 : Segmentations du domaine image. (a) Modes de partition pour un blocs de taille fixe [DWL04], (b) Partition en Quadtree et arbre associé.

Pour adapter la segmentation au contenu de l'image, une solution largement utilisée [PM05, RWB02, Vel05b, Cha05b, DWW+07] est de partitionner le domaine image $\mathcal{D}$ en un arbre quaternaire ou *Quadtree* adaptatif (figure 2.21(b)). Si $\mathcal{D}$ est ramené à un carré défini sur $[0,1]^2$, alors la segmentation de $\mathcal{D}$ en Quadtree est obtenue par division récursive du carré initial en quatre carrés de même taille. Chaque niveau j de l'arbre comporte 2^{j+1} blocs appelés blocs *dyadiques*. Les subdivisions effectuées peuvent être schématisées par un arbre dont chaque nœud possède quatre fils. En termes de coût de codage, 1 bit suffit pour coder une décision de subdiviser un nœud ou pas. Pour une image de dimensions $2^n \times 2^n$, ceci représente au maximum 2^{-n} *bpp* pour coder une structure complète. Pour obtenir une partition adaptée au contenu d'une image, il faut élaguer les branches du Quadtree de sorte que chaque feuille satisfasse un critère donné, par exemple un critère débit distorsion comme expliqué dans le paragraphe suivant.

2.3.2 Création d'un Quadtree adaptatif par optimisation débit-distorsion

Comme nous l'avons vu au premier chapitre, le but d'une optimisation débit-distorsion est de minimiser la distorsion moyenne $\mathbf{D}$ sur l'ensemble des blocs sous la contrainte d'un débit cible $\mathbf{R}_{cible}$ total à ne pas dépasser. Si nous travaillons sous l'hypothèse d'une transformée orthogonale, alors le débit total et la distorsion totale sont la somme des débits et distorsions calculés dans chaque bloc. Ceci simplifie le problème d'allocation de débit.

Supposons que l'image soit partitionnée en N_b blocs et que la partition soit connue. Le débit et la distorsion dans un bloc b_i dépendent du pas de quantification Q_i et des paramètres du modèle géométrique Θ_i. Nous notons Q et Θ l'ensemble des pas de quantification et des paramètres géométriques de tous les blocs. Le problème d'allocation de débit sous contrainte s'écrit alors :

$$(Q^\star, \Theta^\star) = \arg\min_{(Q,\Theta)} \sum_{i=1}^{N_b} \mathbf{D}_i(Q_i, \Theta_i) \quad \backslash \quad \sum_{i=1}^{N_b} \mathbf{R}_i(Q_i, \Theta_i) \leqslant \mathbf{R}_{cible} \tag{2.13}$$

qui équivaut [Ram93b] au problème d'optimisation sans contrainte suivant :

$$(Q^\star, \Theta^\star) = \arg\min_{(Q,\Theta)} \sum_{i=1}^{N_b} \mathbf{J}_i(\lambda) = \arg\min_{(Q,\Theta)} \sum_{i=1}^{N_b} \mathbf{D}_i(Q_i, \Theta_i) + \lambda \mathbf{R}_i(Q_i, \Theta_i) \tag{2.14}$$

Comme on le voit, le multiplicateur lagrangien λ qui règle le compromis débit-distorsion global est le même pour tous les blocs. Etant donné un multiplicateur λ le jeu de paramètres optimal pour un bloc peut être déterminé en calculant les points débit-distorsion $(\mathbf{R}_i, \mathbf{D}_i)$ pour chaque jeu de paramètres possible. Ceci permet de construire la courbe opérationnelle débit-distorsion du bloc. Le point $(\mathbf{R}_i^\star, \mathbf{D}_i^\star)$ pour lequel la pente de la courbe est égale à λ donne le jeu de paramètres optimal recherché. Cette méthode est par exemple utilisé par Velisavljević pour les Directionlets [Vel05b]. D'autres auteurs, comme Le Pennec et Mallat [PM05] ou Chappelier [Cha05b] utilisent une approximation bas débit qui permet d'exprimer le λ en fonction du pas de quantification Q_i uniquement. Ceci accélère le procédé car seuls les jeux de paramètres géométriques candidats doivent alors être testés.

Supposons maintenant que l'on souhaite segmenter l'image en un Quadtree adaptatif de manière à minimiser le coût lagrangien total $\mathbf{J}(\lambda) = \mathbf{D} + \lambda\mathbf{R}$ pour un λ donné. En plus du pas de quantification et de la géométrie à l'intérieur d'un bloc, il faut ici déterminer la coupe optimale à réaliser dans le Quadtree. Pour ce faire, la méthode la plus utilisée [PM05, Cha05b, Vel05b, RWB02, DWW+07] se déroule comme suit. Le point débit-distorsion $(\mathbf{R}_i, \mathbf{D}_i)$ associé au lagragien λ est calculé pour chaque bloc dyadique à tous les niveaux du Quadtree (en pratique, les auteurs [PM05, Cha05b] se limitent souvent à une profondeur correspondant à des blocs 4×4) et le coût lagrangien $\mathbf{J}_i(\lambda)$ est retenu. Ensuite, un algorithme de type « top-down » est mis en place pour élaguer les branches de l'arbre. On débute au niveau de profondeur maximal J de l'arbre. Tout bloc dyadique b_i au niveau de profondeur $(J-1)$ peut être découpé en 4 blocs notés b_l au niveau J : $b_i = \cup_l b_l$. La décision de couper la branche correspondante du Quadtree est prise si :

$$\mathbf{J}_i(\lambda) \leqslant \sum_l \mathbf{J}_l(\lambda) \tag{2.15}$$

En poursuivant ces décisions le long de chaque branche du Quadtree, on obtient finalement la segmentation optimale pour le multiplicateur λ considéré.

Souvent, le lagrangien $\lambda^\star$ donnant la distorsion minimale sur l'ensemble de l'image pour le débit $\mathbf{R}_{cible}$ n'est pas connu d'avance. Pour le déterminer, il faut alors faire une recherche sur λ, par exemple une recherche dichotomique comme expliqué dans [Ram93b]. Dans [DWW+07], Ding et al. ont une solution pour contourner le problème. En effet, leur approche basée sur le schéma lifting permet d'adopter le codeur EBCOT pour encoder les coefficients d'ondelettes. Ils émettent alors l'hypothèse que leur méthode produit simplement une translation des courbes opérationnelles obtenues avec EBCOT. En codant l'image avec EBCOT avant d'appliquer leur propre codeur, cette stratégie leur permet donc de connaître λ^*.

2.3.3 Gestion des effets de bords

Une image est définie sur un support borné. Lorsqu'une fonction de base intersecte les bords du domaine image, il est donc nécessaire de la modifier pour obtenir une base orthonormée. Typiquement, ceci se fait en considérant une extension périodique ou symétrique de l'image. En découpant l'image en blocs, le problème apparaît aux bords de chaque bloc. S'il n'est pas pris en compte correctement, de nouvelles discontinuités désagréables à l'œil peuvent apparaître lors d'une approximation de l'image.

Dans le cas des Directionlets [Vel05b], l'effet de bords est clairement indiqué comme une limite de l'approche dans le cas des images naturelles. En effet, l'auteur utilise une extension symétrique le long des co-lignes, mais ceci n'empêche pas l'apparition d'artefacts.

Dans l'approche proposée par Taubman et Zakhor [TZ94b], un bloc de l'image est déformé en parallélogramme avant d'être décomposé dans une base d'ondelettes 2D standard (paragraphe 2.2.4.2). L'ondelette choisie est une ondelette 9/9 de Adelson et al. [ASH87]. Les blocs déformés sont transformés les uns indépendamment des autres en utilisant une extension symétrique aux bords des lignes et des colonnes. Le fait d'avoir un support parallélogramme rend la tâche un peu plus complexe car certaines lignes ou colonnes sont trop courtes pour générer une extension symétrique suffisante. Pour ces lignes ou colonnes particulières, l'ondelette de Adelson est simplement remplacée par l'ondelette de Haar qui ne nécessite pas d'extension. Pour limiter les effets de blocs, les auteurs proposent tout simplement d'utiliser de grands blocs de taille 64×64. Un post-traitement est réalisé au décodage pour lisser les discontinuités aux frontières des blocs. On peut se demander cependant si la géométrie dans des blocs aussi grands peut être correctement capturée par des filtrages rectilignes.

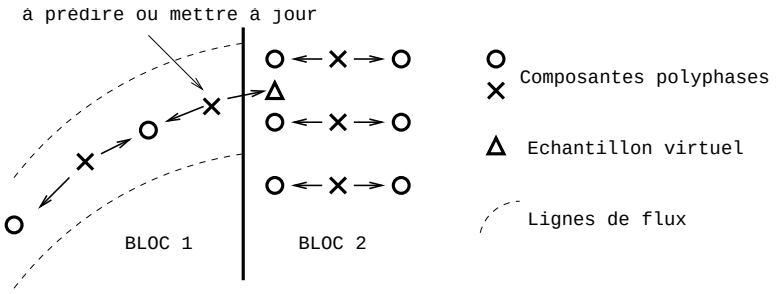


Fig. 2.22 : Gestion des bords pour les Bandelettes première génération. L'échantillon virtuel est obtenu en interpolant les ronds dans la colonne.

Les mêmes questions se posent dans l'approche par Bandelettes proposée par Le Pennec et Mallat [PM05]. Pour mieux comprendre comment les auteurs gèrent le filtrage aux bords, considérons un bloc d'origine de l'image où le flux est parallèle verticalement. Dans le domaine d'origine $\mathcal{D}$, les échantillons donnant les valeurs du bloc déformé peuvent être placés le long des courbes de flux. Les auteurs proposent alors d'exploiter le schéma en lifting pour gérer les problèmes aux bords. Prenons la configuration de la figure 2.22 où l'on souhaite prédire ou mettre à jour la valeur d'un échantillon situé en

bord de bloc. Cet échantillon a un voisin à gauche dans son bloc le long de la courbe de flux, mais pas de voisin à droite. Les auteurs proposent alors de créer un échantillon virtuel à droite dont la valeur est obtenue en interpolant les valeurs de deux échantillons du bloc voisin. Ceci revient à étendre le signal le long de la courbe de flux. Notons que l'extension le long de la courbe de flux est limitée à un échantillon pour assurer la réversibilité, ce qui impose des contraintes sur l'ondelette à utiliser aux bords. Ainsi, à l'intérieur d'un bloc les auteurs utilisent l'ondelette de Daubechies 9/7 [ABMD92] à 4 moments nuls tandis qu'aux bords ils utilisent une ondelette à 2 moments nuls. Même si ce changement implique une perte d'orthogonalité de la base de bandelettes aux frontières de blocs, la proposition des auteurs permet néanmoins d'assurer une certaine continuité du filtrage aux frontières. Notons que dans le cas des Bandelettes seconde génération, la transformée de Alpert ne nécessite pas d'extension aux bords des blocs, ce qui permet de conserver partout la propriété d'orthogonalité.

Pour éviter les problèmes aux bords de blocs, l'idéal est qu'une continuité naturelle existe dans le modèle géométrique lorsque l'on passe d'un bloc à l'autre. En particulier, arrêtons nous sur le schéma de lifting directionnel de Wang et al. [WZVS06]. La géométrie dans un bloc est modélisée par deux directions de filtrage repérées par θ_v et θ_h. L'ensemble des orientations permises est choisi de sorte qu'un pixel ne peut être mis en correspondance qu'avec un point ayant une précision maximale au demi pixel dans les colonnes ou les lignes adjacentes (figure 2.23(a)). En prolongeant les segments de flux d'un bloc à l'autre, deux réseaux de lignes de flux linéaires par parties sont construites : l'un globalement horizontal, l'autre globalement vertical. Des règles sont définies pour que chaque pixel de l'image appartienne à une et une seule courbe horizontale et verticale. Un filtrage continu globalement horizontal puis globalement vertical peut alors être effectué d'un bord à l'autre de l'image. Ceci permet d'éviter les difficultés aux frontières de blocs. Remarquons que la construction des deux réseaux de lignes n'est possible que si les orientations θ_v et θ_h sont discrétisées pour permettre des correspondances avec une précision maximale au demi-pixel.

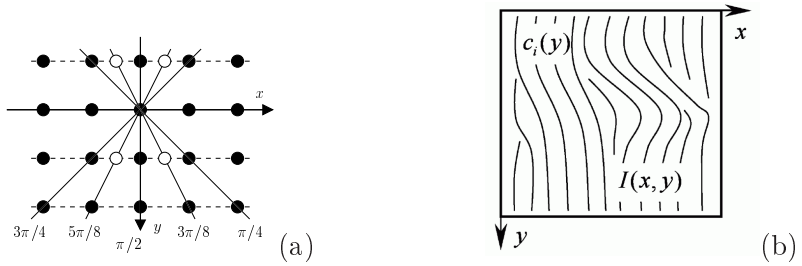


Fig. 2.23 : Méthode de Wang et al. [WZVS06]. (a) Orientations de filtrage vertical permises, (b) Un réseau de lignes de flux globalement verticales.

2.3.4 Maillage 2D

Le maillage 2D est un modèle en marge par rapport aux modèles vus précédemment. En compression d'images fixes, il est souvent utilisé pour bâtir des approximations

globales des images par éléments finis comme expliqué dans les paragraphes suivants. Notons que dans nos travaux (chapitres 4 et 5), nous avons fait un usage différent du maillage 2D : l'idée est de l'utiliser comme modèle déformable afin de représenter des déformations du contenu de l'image à la manière des Bandelettes ou d'une compensation en mouvement. Cette déformation nous permet d'adapter le contenu de l'image à la transformée par ondelettes standard (horizontale-verticale). Dans la suite, nous donnons quelques définitions relatives au maillage et présentons l'usage qui en est fait classiquement dans le cadre de l'image fixe.

2.3.4.1 Définitions

Un maillage $\mathcal{M}$ est un ensemble de sommets, d'arêtes et de facettes. Il est caractérisé par deux types d'information : sa *géométrie*, c'est-à-dire les positions de ses sommets dans un espace de dimension d et sa *topologie*, c'est-à-dire les relations de connectivité qui lient les différents éléments entre eux. Dans ce manuscrit, nous considérerons essentiellement des maillages de géométrie 2D avec bords dont la superficie couvre le domaine image $\mathcal{D}$. Un maillage est dit *régulier* si tous ses sommets internes ont la même *valence*, c'est-à-dire le même nombre d'arêtes incidentes. Il est dit *irrégulier* dans le cas contraire.

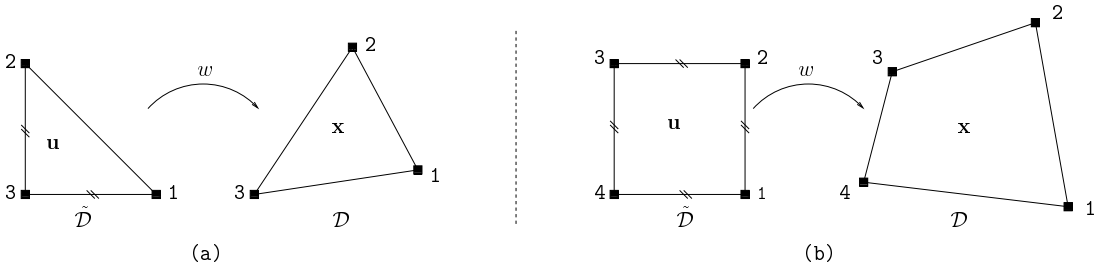


Fig. 2.24 : L'élément maître pour (a) un maillage triangulaire et (b) un maillage quadrangulaire [WL94].

Les facettes les plus rencontrées dans la littérature sont les triangles et les quadrilatères. Toute facette de géométrie quelconque peut être mise en correspondance avec une facette de géométrie fixe, appelée *élément maître* dans [WL94, LW95]. Cet élément est défini sur un domaine appelé *domaine maître* $\tilde{\mathcal{D}}$. L'élément maître est simple (voir figure 2.24) et permet de définir facilement certaines fonctions (noyau de représentation, fonction d'interpolation) dans le domaine maître. La déformation de la maille de $\tilde{\mathcal{D}}$ à $\mathcal{D}$ permet de définir une transformation spatiale w. Notons $\mathbf{u} = (u, v)$ un point dans $\tilde{\mathcal{D}}$ et $\mathbf{x} = (x, y)$ son correspondant par w dans $\mathcal{D}$: $\mathbf{x} = w(\mathbf{u})$. Nous définissons le *jacobien* $J_w(\mathbf{u})$ de la déformation en un point $\mathbf{u}$ comme :

$$J_w(\mathbf{u}) = \begin{vmatrix} \frac{\partial x}{\partial u} & \frac{\partial y}{\partial u} \\ \frac{\partial x}{\partial v} & \frac{\partial y}{\partial v} \end{vmatrix} \tag{2.16}$$

Ce jacobien est important car il permet de caractériser des cas de mailles dites *dégénérées* qui nuisent à la robustesse des algorithmes. Des exemples de mailles quadrangulaires dégénérées sont illustrées figure 2.25. Une facette est dite *conforme* si le jacobien est supérieur à 0 en tout point de la facette.

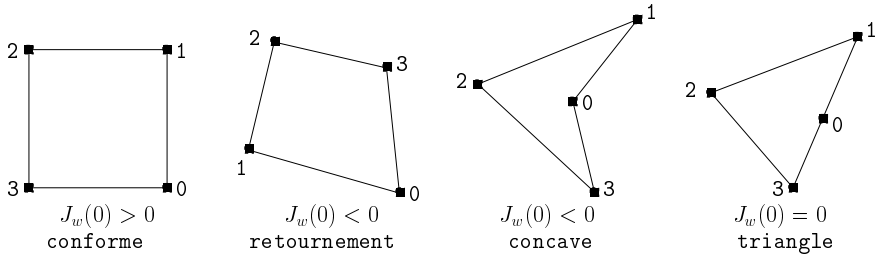


Fig. 2.25 : Une maille quadrangulaire conforme et les trois cas de dégénérescences possibles. Pour chaque cas, la dégénérescence est détectée au nœud d'indice 0.

2.3.4.2 Approximation par éléments finis

En codage d'images fixes, le maillage 2D est le plus souvent utilisé comme une *grille d'échantillonnage* pour une approximation de l'image par éléments finis [LW95, LLS99, MPL00b, DDI06]. Une telle approximation se construit en associant une intensité à chaque sommet du maillage. Les valeurs sur la grille des pixels sont ensuite calculées en interpolant les intensités des nœuds, par exemple en définissant une fonction de forme dans le domaine maître. L'image est alors représentée par un maillage 2D et par un ensemble d'intensités. Plus la géométrie du maillage reflète la géométrie de l'image, plus l'approximation est fine. Le maillage est donc bien un modèle de géométrie.

En termes d'approximation linéaire, on peut montrer [PM05] que si I est $\mathcal{C}^2\backslash\mathcal{C}^2$, alors l'approximation $\tilde{I}_M$ avec des éléments finis linéaires sur M triangles vérifie le taux de décroissance optimal :

$$\|I - \tilde{I}_M\|^2 \leqslant K \cdot M^{-2},$$

où K est une constante qui ne dépend que de la fonction I. Il est même possible d'obtenir un exposant $M^{-\alpha}$ pour des images $\mathcal{C}^\alpha\backslash\mathcal{C}^\alpha$ à condition d'utiliser des éléments finis d'ordre plus élevé. Cependant, ces résultats d'approximation supposent qu'aucune contrainte n'est émise sur la connectivité du maillage.

Or, pour une application en compression, cette information de connectivité a un coût au même titre que la géométrie du maillage. En général, lorsqu'un maillage est régulier, la valence est fixée arbitrairement et donc le coût de la connectivité est nul. A contrario, pour un maillage irrégulier les relations d'incidence ne sont pas connues a priori et peuvent varier fortement en fonction de la géométrie locale de l'image. Un tel maillage permet une meilleure adaptation géométrique mais le coût de sa connectivité devient rapidement prohibitif. Notons cependant le cas particulier de la triangulation

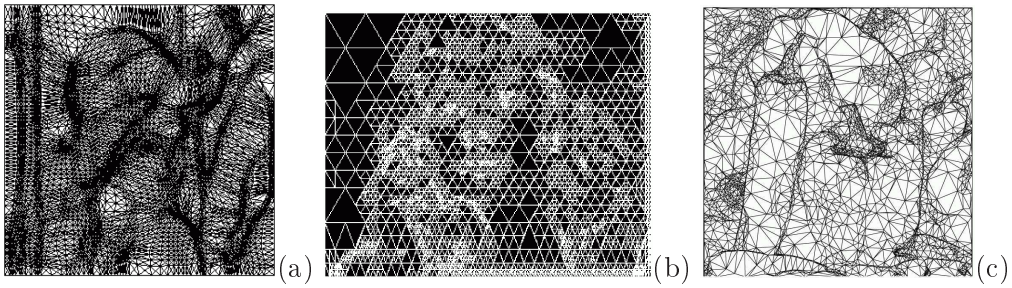


Fig. 2.26 : Géométries approchées par des maillages 2D. (a) Maillage régulier sur *Lena* [TV91], (b) Maillage Quadtree à géométrie fixe sur *Suzie* [MPL00b], (c) Triangulation de Delaunay sur *Peppers* [DDI06].

de Delaunay [DDI06], maillage irrégulier mais défini complètement par la seule connaissance de sa géométrie (figure 2.26(c)). Pour obtenir un compromis entre adaptivité et coût de connectivité, il est possible de construire des maillages semi-réguliers. Il s'agit de créer tout d'abord un maillage « grossier » irrégulier puis de *raffiner* ce maillage régulièrement.

Raffiner une facette triangulaire ou quadrangulaire revient généralement à la subdiviser en 4 facettes de même forme. Ceci se fait en créant de nouveaux sommets sur chaque arête (voir figure 2.27). On observe que la subdivision récursive d'une facette triangulaire crée un maillage régulier dont les sommets ont une valence 6. De même, la subdivision d'un quadrilatère quelconque crée un maillage régulier avec des sommets de valence 4. En vision par ordinateur, les surfaces 3D définies sur de telles grilles sont appelées *surfaces de subdivision*. Dans la suite, un maillage régulier fera toujours référence à un maillage de valence 6 s'il est triangulaire ou 4 s'il est quadrangulaire.

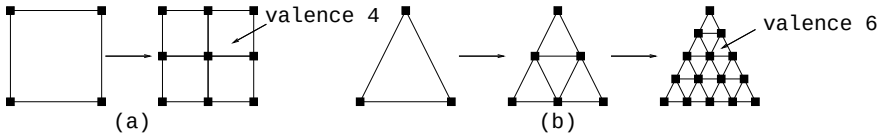


Fig. 2.27 : Subdivision d'une facette (a) quadrangulaire et (b) triangulaire.

Subdiviser les facettes d'un maillage améliore son niveau de détail et donc sa capacité à capturer les singularités géométriques. Comme une subdivision génère 4 nouvelles facettes, il est possible de représenter les subdivisions par un Quadtree. En élaguant les branches de l'arbre, on obtient alors un « maillage Quadtree » où la densité des nœuds est fonction de la géométrie locale (figure 2.26(b)). Notons que si un maillage est régulier par parties alors il est possible d'inverser le processus de subdivision pour obtenir des approximation multi-échelles de la géométrie et de l'image. Cette propriété est exploitée par les ondelettes dites géométriques.

2.3.4.3 Ondelettes géométriques sur maillage

Supposons que la géométrie d'une image soit modélisée à l'aide d'un maillage 2D régulier. Des intensités sont associées à chaque nœud pour construire une approximation de l'image. Avant d'encoder la géométrie du maillage et les intensités aux nœuds, il est possible de les décomposer dans une base d'ondelettes. En particulier, les nœuds d'un maillage régulier (quadrangulaire ou triangulaire) peuvent être mis en correspondance avec les nœuds d'une grille carrée uniforme dans le domaine maître. Sur cette grille uniforme, des bases d'ondelettes séparables 2D peuvent être définies sans ambiguïté comme sur une grille de pixels. A chaque nœud i du domaine maître peut être associée une position 2D (x_i, y_i) dans le domaine image et une intensité I_i. Chaque ensemble $\{x_i\}_i$, $\{y_i\}_i$ et $\{I_i\}_i$ regroupe les valeurs d'une fonction 2D discrète définie sur le maillage maître. Ces valeurs peuvent être décomposées avec la base d'ondelettes choisie comme toute fonction définie sur une grille de pixels.

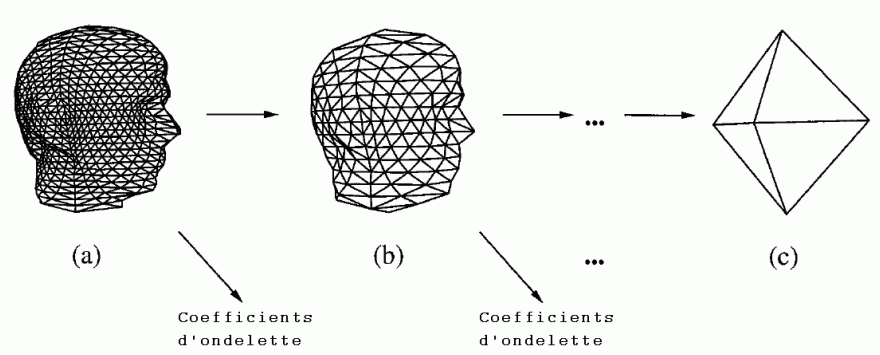


Fig. 2.28 : Décomposition multi-résolutions d'un maillage 3D. D'après [LDW97].

Lorsque le maillage est triangulaire, d'autres bases d'ondelettes non séparables ont été introduites dans la communauté de vision par ordinateur. Ainsi, c'est en 1987 que Loop [Loo87a] introduit la notion de *surfaces de subdivision*. Partant de quelques facettes triangulaires approximant grossièrement un objet 3D, il montre qu'il est possible de créer une surface 3D de plus en plus lisse en subdivisant chaque triangle de façon récursive comme expliqué au paragraphe précédent. A chaque étape, la position dans l'espace 3D des nouveaux sommets créés est interpolée à partir de positions des sommets voisins déjà existants. Ceci se fait en utilisant des fonctions B-spline définies sur le maillage parent. L'étude de Loop ressemble donc très fortement au processus de *synthèse* d'une fonction tridimensionnelle dont la régularité est donnée par la B-spline. En s'appuyant sur ces travaux et sur le schéma en Lifting de Sweldens, Lounsbery et al. [LDW97] mettent en forme l'*analyse* multi-résolutions des surfaces 3D (figure 2.28). A chaque étape de décomposition, la position d'un sommet au niveau de résolution j est prédite uniquement avec les positions des sommets de niveau $(j-1)$ (figure 2.29). La différence entre la position d'origine et la position prédite donne un vecteur d'ondelettes ou détail 3D. Les positions des sommets de niveau $(j-1)$ sont ensuite mises à jour et

les arêtes de niveau j sont supprimées. La prédiction d'une position peut se faire en utilisant uniquement les positions des deux sommets de niveau j qui lui sont incidents. C'est le schéma adopté par l'ondelette Midpoint par exemple. La prédiction peut aussi se faire avec des sommets plus éloignés comme pour l'ondelette Butterfly [DLG90] ou l'ondelette de Loop [Loo87b]. Dans un contexte de compression d'images fixes où la géométrie est modélisée par un maillage triangulaire régulier, toutes ces ondelettes peuvent être utilisées pour décomposer les valeurs discrètes $\{x_i\}_i$, $\{y_i\}_i$ et $\{I_i\}_i$.

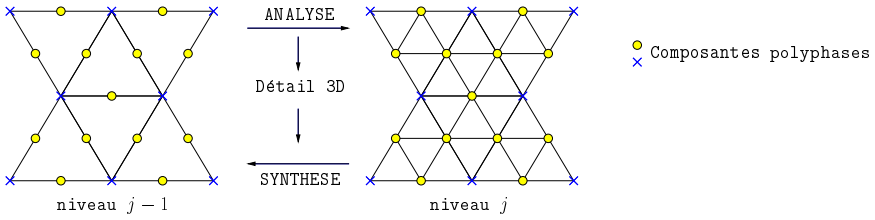


Fig. 2.29 : Procédé d'analyse et de synthèse sur une portion d'un maillage triangulaire régulier. Dans le cas de l'ondelette Butterfly, le point 3D associé à l'échantillon rond central est prédit avec les points 3D associées à tous les échantillons croix.

Lorsque le maillage est semi-régulier, la décomposition multi-résolutions reste possible. Le niveau de décomposition maximal est cependant limité par la taille des zones régulières et les fonctions de base doivent être modifiées à la frontière entre deux zones.

2.3.4.4 Estimation du maillage

Dans ce paragraphe, nous décrivons quelques méthodes antérieures qui ont été mises en œuvre pour calculer un maillage 2D permettant d'approximer une image. Notons que des méthodes indépendantes ont été développées pour calculer des approximations d'une surface 3D quelconque. Nous pouvons par exemple orienter le lecteur vers les travaux de Agarwal et Suri [AS98], Hoppe et al. [HDD+93, Hop96], Garland et Heckbert [GH97] ou Lindstorm et Turk [LT98]. Dans le contexte de la compression d'images fixes, de nombreux travaux ont également été proposés. Considérons tout d'abord le cas où la connectivité du maillage n'est pas contrainte.

Connectivité quelconque. Partant d'un maillage triangulaire régulier dense, Lechat [Lec99b] puis Brangoulo [Bra05b] mettent en œuvre un algorithme itératif pour aboutir à un maillage triangulaire adaptatif à connectivité quelconque. Chaque itération comprend des fusions de polygones, des permutations de diagonales et une optimisation des positions et niveaux de gris aux nœuds. Ces opérations sont guidées par l'erreur quadratique d'approximation. Pour coder la connectivité du maillage obtenu, des algorithmes performants existent, comme ceux de Deering [Dee95], Taubin et Rossignac [TR98] ou Touma et Gotsman [TG98].

Dans [FL96], Le Floch et Labit abordent le problème d'approximation comme un problème de sous-échantillonnage d'image avec un nombre fixé d'échantillons. Leur tech-

nique est en marge des précédentes car l'approximation de l'image est ici effectuée en utilisant des fonctions d'interpolations ayant un support isotrope (circulaire) et centrées sur les nouveaux échantillons. Il n'y a pas réellement de maillage donc pas de connectivité à transmettre.

Plus récemment, Demaret et al. [DDI06] proposent de construire une triangulation de Delaunay à partir d'un petit nombre de pixels « signifiants » choisis de façon adaptative qui deviennent les nœuds du maillage. L'algorithme sélectionnant les pixels signifiants est un algorithme glouton. Il s'agit d'éliminer un par un les pixels les moins signifiants en se basant sur l'erreur quadratique d'approximation produite par la suppression de chaque pixel et paire de pixels. Notons que le calcul d'une triangulation de Delaunay après suppression d'un sommet peut se faire rapidement à partir de la triangulation de Delaunay courante. Comme une triangulation de Delaunay peut être reconstruite uniquement à l'aide des positions des sommets, aucune connectivité ne doit être transmise.

Connectivité régulière. Considérons maintenant le cas où la connectivité du maillage est contrainte pour limiter son coût. La contrainte la plus forte pour un coût nul est la régularité sur tout le domaine image. Terzopoulos et Vasilescu [TV91] par exemple proposent de couvrir le domaine image avec un maillage triangulaire régulier dont chaque arête est assimilée à un ressort. Le nombre de nœuds est fixé a priori et la constante de chaque ressort dépend d'une observation locale liée au gradient. L'ensemble de tous les ressorts forme un système physique dont l'état d'équilibre est recherché récursivement.

Dans [LW95], Lee et Wang proposent quant à eux de minimiser une énergie d'interpolation. Pour un ensemble donné de positions des nœuds, l'énergie d'interpolation est l'erreur quadratique totale d'approximation dans le domaine image. Les auteurs préfèrent exprimer cette énergie dans le domaine maître où le maillage est fixe et les fonctions de forme bien définies. L'énergie est minimisée en annulant sa dérivée. Une énergie ressort appelée énergie déformation (voir Wang et Lee [WL94] et chapitre 3) est ajoutée pour contrôler la déformation du maillage.

Les travaux de Jansen et al. [JCLB01] sont en marge des deux travaux précédents. Les auteurs considèrent l'image comme une surface 3D et adoptent une approche « bottom-up » pour construire une surface de subdivision adaptée. A chaque étape, de nouveaux nœuds sont positionnés au centre des arêtes 3D puis sont déplacés dans la direction de la normale à la surface en adoptant une heuristique propre aux « Normal Meshes » de Guskov et al. [GVSS00].

Connectivité semi-régulière. Pour atteindre une certaine qualité d'approximation dans une région donnée de l'image, le nombre de nœuds (et de facettes) nécessaires dépend du contenu de la région. Une zone homogène nécessitera par exemple moins de nœuds qu'une région présentant un contour. Lorsque de fortes disparités existent dans l'image, il peut donc être avantageux de construire un maillage semi-régulier. Dans les travaux de Lee et Wang [LW95], une approche « bottom-up » adaptive est ainsi proposée. A chaque étape, une maille est subdivisée dans les deux cas suivants : soit

elle contient une zone homogène mal approximée (au sens de l'erreur d'interpolation), soit elle contient un contour. Si la maille contient une zone homogène bien approximée ou une zone texturée, elle n'est pas subdivisée. Le choix de ne pas subdiviser la maille dans une région texturée vient de l'observation faite par les auteurs que *les erreurs dans les régions texturées ne détériorent pas significativement la perception visuelle*. Pour classifier les régions, Lee et Wang utilisent des descripteurs statistiques comme ceux présentés par Vaisey et Gersho [VG92]. Les décisions de subdiviser ou non une maille sont enregistrées dans une structure en Quadtree.

De la même façon, Lechat et al. [LLS99] construisent un maillage de façon hiérarchique. A chaque étape, la décision de subdiviser ou non une maille dépend de l'invariance locale, du contraste et de l'erreur quadratique de reconstruction. Les valeurs et positions nodales sont également mises à jour.

L'approche proposée par Marquant et al. [MPL00b] se distingue des approches précédentes car elle prend en compte le coût de la connectivité (arbre de décisions). En outre, dans cette méthode les positions des nœuds à un niveau hiérarchique donné sont fixes et connues a priori de sorte que seul l'arbre de décisions doit être transmis (aucune géométrie ne doit être transmise).

Citons enfin la démarche proposée par Brangoulo [Bra05b] qui s'articule en trois temps. Dans un premier temps, une carte de saillance de l'image est construite en s'appuyant sur les sous-bandes ondelettes [LLMD06]. Cette carte permet d'extraire un ensemble de points saillants. Dans un second temps, une triangulation de Delaunay est créée en considérant chaque point saillant comme un sommet. Enfin, chaque maille de la triangulation est raffinée régulièrement en fonction de son contenu à l'aide d'un processus de subdivision proche de celui de Lechat et al. [LLS99].

2.4 Compression

2.4.1 Codage des sous-bandes

Dans les sections précédentes, nous avons présenté différentes représentations adaptatives d'une image basées sur la déformation de l'ondelette. Leur résultat en termes d'approximation non linéaire a été précisé lorsqu'il était connu. Au premier chapitre, nous avons souligné que ce résultat, souvent établi théoriquement pour une classe d'images particulières, ne conditionne pas totalement la performance finale en compression sur des images naturelles. Ceci a été illustré avec l'exemple de la transformée en ondelettes standard : elle présente un résultat d'approximation sous-optimal mais les propriétés statistiques des sous-bandes d'ondelettes permettent à des codeurs comme EZW, SPIHT ou EBCOT d'exploiter efficacement les résidus de corrélation. La question est de savoir si de tels codeurs peuvent être appliqués aux sous-bandes générées par les nouvelles représentations.

Dans le cas des schémas de lifting directionnel [DWL04, DWW$^+$07, WZVS06, Cha05b], les sous-bandes générées ont la même forme que dans une décomposition en ondelettes dyadique. Elles peuvent donc être encodées avec un codeur ondelettes sans modification avancée. Ceci permet en particulier aux auteurs de comparer leur technique avec

JPEG2000. Dans le cas des Directionlets [VBLVD06], la décomposition génère des sous-bandes dont la taille n'est pas dyadique. Ceci est dû à l'utilisation d'un niveau de décomposition différent dans les deux directions données par la lattice Λ. L'auteur propose alors une extension de l'algorithme EZW qui modifie les relations entre un nœud parent et sa descendance dans un arbre de zéros. Le nombre d'enfants dépend ainsi du ratio d'aspect choisi. En outre, les enfants sont situés sur une version sous-échantillonnée de la lattice Λ.

Dans le cas où la transformation en ondelettes se fait sur des blocs déformés [TZ94b, PM05], les sous-bandes générées ne sont pas définies sur des grilles dyadiques carrées après décomposition. Dans [TZ94b], Taubman et Zakhor utilisent un codage DPCM (« Differential Pulse Code Modulation ») pour la sous-bande basse fréquence puis encodent les autres sous-bandes à l'aide d'un codage à longueur variable. Notons qu'il serait cependant possible de replacer les échantillons sur des grilles dyadiques après décomposition par déformation inverse pour utiliser un codeur ondelettes comme JPEG2000. Dans le cas des Bandelettes, le procédé de Bandelettisation décrit plus haut rend la structure des coefficients de Bandelettes plus complexe à modéliser que celle des coefficients d'ondelette. Ainsi, dans [PM05] Le Pennec et Mallat choisissent de coder les sous-bandes quantifiées de bandelettes à l'aide d'un codeur arithmétique sans incorporer les contextes propres à JPEG2000. La méthode est alors comparée à un codeur ondelettes équivalent. Dans [Pey05b], Peyré utilise également un codage arithmétique des coefficients quantifiés et souligne la complexité de construire des contextes adaptés à la structure des coefficients en bandelettes. Il fournit néanmoins une comparaison de la transformée en Bandelettes seconde génération par rapport à JPEG2000 en termes de PSNR et ne note pas de gain significatif sur ce plan.

Enfin, notons que des applications à la compression s'appuyant sur la transformée en Contourlets présentée en section 2.1 ont aussi été étudiées. Rappelons que cette transformée est non adaptative et consiste à analyser chaque sous-bande haute fréquence d'une pyramide Laplacienne selon plusieurs directions en combinant filtres en éventail et sous-échantillonnage directionnel sur lattices. Les travaux d'origine n'offrent pas d'application en compression. Plus récemment, Eslami et Radha [ER04] ont proposé d'effectuer le filtrage directionnel sur les sous-bandes obtenues avec une décomposition classique en ondelettes. La décomposition est non redondante et les sous-bandes peuvent être encodée à l'aide d'un algorithme similaire à SPIHT. Les résultats numériques de compression sont en-dessous de ceux obtenus avec les ondelettes standards sur l'ensemble de la gamme de débits. Chappelier et Guillemot [CGM04b] proposent quant à eux de n'effectuer la transformée en Contourlet que sur un nombre limité de niveaux de la pyramide Laplacienne, puis de poursuivre avec une décomposition standard. Ceci permet de contrôler la redondance de la transformée. Les auteurs réalisent une série de tests de compression en codant les sous-bandes avec un codeur de type EZBC adapté à la transformée en Contourlets. Une optimisation similaire à celle effectuée dans JPEG2000 est réalisée. Ils notent un léger gain objectif à bas débit pour des images possédant des caractéristiques directionnelles. Les performances chutent à haut débit du fait de la redondance de la transformée.

2.4.2 Remarques sur la « scalabilité »

Nous avons vu au chapitre 1 que la scalabilité est un enjeu important des recherches en compression. JPEG2000 est capable de générer de façon très performante un flux « scalable » spatialement et en SNR. Il est essentiel que les nouveaux codeurs conservent ces propriétés. Pourtant, la problématique de compression scalable est peu mentionnée dans les articles consacrés aux ondelettes seconde génération. Les paramètres sont en général optimisés pour atteindre un débit cible et non pour créer un flux scalable. Intéressons nous aux capacités de ces méthodes en termes de scalabilité.

Dans un premier temps considérons que la *scalabilité géométrique* n'est pas un prérequis. La scalabilité SNR ne pose a priori pas de problème car les codeurs de sous-bandes proposés permettent quasiment tous un codage progressif. Egalement, la majorité des représentations présentées précédemment sont multi-échelles et se prêtent donc bien à un codage scalable spatialement. Remarquons simplement que les sous-bandes de basses fréquences générées par les transformées en Directionlets [Vel05b] ne respectent pas le ratio d'aspect de l'image d'origine. Ceci est dû au sous-échantillonnage sur une lattice[3] et empêche une scalabilité spatiale naturelle.

Dans un cadre de codage complètement scalable, toutes les informations doivent être encodées de manière scalable et cela inclut la géométrie. Au niveau du décodage, la qualité et la précision de la géométrie doivent être adaptées à la résolution spatiale et à la distorsion visuelle. La plupart des études citées précédemment ne prennent pas cet élément en considération. Ainsi, l'optimisation Lagrangienne décrite plus haut estime une géométrie dans un bloc pour une résolution spatiale (la résolution du bloc) et SNR (le lagrangien λ) données. Pour représenter la géométrie dans un bloc sur plusieurs niveaux de résolution, il faut définir et calculer de nouveaux paramètres pour chaque niveau de décomposition ondelettes. Le plus souvent, pour limiter le coût de la géométrie, les auteurs calculent des paramètres au niveau de résolution le plus fin et ré-utilisent ces mêmes paramètres pour les niveaux suivants. Il n'y a donc pas de scalabilité en géométrie. Dans certains cas particuliers comme la décomposition en Bandelettes seconde génération ou la décomposition en Wedgelets, des paramètres géométriques sont cependant calculés sur plusieurs niveaux de résolution, permettant ainsi la scalabilité géométrique.

Pour ne pas avoir à coder la géométrie sur plusieurs niveaux de résolution, une solution consiste à la coder au niveau de résolution le plus fin puis à la décoder avec perte. Néanmoins, cette solution convient mal aux techniques basées sur un filtrage directionnel dans le domaine image. En effet, lors de la décomposition, les coefficients d'ondelettes à chaque niveau sont déterminés en opérant un filtrage dans une orientation donnée. Supposons que cette orientation soit décodée avec une légère perte. Lors de la synthèse, cette perte aura un impact d'autant plus important sur la qualité de l'image reconstruite que le nombre de niveaux de décomposition sera élevé, les erreurs de reconstruction se cumulant à chaque niveau.

En revanche, la solution de décoder la géométrie avec perte convient bien aux méthodes basées sur des déformations de blocs. En effet, dans ces méthodes, la géométrie

[3] Le ratio d'aspect obtenu après une décomposition est donné par les vecteurs de la matrice $\mathbf{M}_\Lambda$.

est extraite lors de la déformation des blocs et n'est pas utilisée lors de la décomposition en ondelettes qui se fait selon les directions horizontale et verticale. Les blocs déformés peuvent donc être reconstruits sans perte. Les pertes sur la géométrie n'ont d'impact qu'au moment de la déformation inverse des blocs. Cette propriété est exploitée par Le Pennec et Mallat dans la toute première construction des Bandelettes [PM00]. Les auteurs notent qu'à bas débits une trop large part de la bande passante est affectée aux courbes géométriques. Ils proposent donc de les décoder avec perte pour pouvoir affecter plus de débits aux coefficients de Bandelettes. Les auteurs remarquent que *cette perte a un impact fort sur le PSNR mais n'affecte guère la qualité visuelle des blocs reconstruits*. On notera cependant que les Bandelettes traitent les blocs d'une image indépendamment. Lorsqu'un contour traverse la frontière entre deux blocs, une perte géométrique dans chaque bloc provoque donc des ruptures de continuité visibles à l'œil.

Pour finir, observons que la problématique de scalabilité se simplifie grandement lorsque l'image est ré-échantillonnée par un maillage. En effet, comme nous l'avons vu au cours de ce chapitre, un maillage qu'il soit régulier ou irrégulier peut être décomposé sur plusieurs niveaux de résolution. Ceci permet de créer un flux scalable spatialement pour les intensités et les positions des sommets.

Conclusion

Dans ce chapitre, nous nous sommes intéressés à différents outils antérieurs permettant d'intégrer la dimension géométrique au noyau de représentation. Nous avons distingué les noyaux fixes des noyaux adaptatifs. Les noyaux fixes ont une géométrie indépendante de l'image à analyser. Ils peuvent donc représenter l'image sans paramètre d'adaptation annexe. Les noyaux adaptatifs quant à eux sont formés à l'aide de paramètres géométriques annexes. Une façon de créer un noyau adaptatif est de déformer l'ondelette séparable en fonction du flux géométrique local.

Différentes représentations adaptatives basées sur des modélisations locales du flux géométrique ont donc été présentées. Le filtrage sur une lattice permet une décomposition directionnelle de même complexité que la décomposition par ondelettes séparables classique. Le lifting orienté autorise la création de nouveaux échantillons interpolés et reste sans perte. La déformation de bloc autorise les pertes numériques pour permettre un suivi précis des lignes de flux. Nous avons vu également que le ratio d'aspect de l'ondelette pouvait être modulé en jouant sur les niveaux de décomposition dans les deux directions de filtrage.

Nous nous sommes ensuite intéressés à des modélisations globales de la géométrie. La structure en Quadtree a été mise en avant ainsi que la manière de la construire par optimisation débit-distorsion. Nous avons ensuite fait un focus sur la représentation par maillage qui permet de construire une approximation par éléments finis des images. Lorsque le maillage est régulier, cette approximation peut être représentée sur différents niveaux de résolution, par exemple en adaptant l'ondelette à la nouvelle grille d'échantillonnage.

En termes d'approximation non linéaire, la plupart des approches citées dans ce chapitre apportent un gain significatif par rapport à l'approximation en ondelettes. Dans un cadre de compression, le gain dépend beaucoup de l'approche et de la gamme de débits ciblée. Pour être juste, la comparaison doit se faire avec le codeur de l'état de l'art qui est JPEG2000. D'une manière générale, les codeurs basés sur une transformée adaptative montrent des gains visuels intéressants lorsque l'image possède des caractéristiques géométriques. Ce gain visuel est surtout significatif dans les bas débits jusqu'à 0.5 bpp où l'on note une réduction des effets de pixellisation et de « ringing » autour des contours. Notons que les courbes de PSNR ne traduisent pas forcément ce résultat visuel. Dans les hauts débits, lorsque la transformée est sans perte les performances objectives de JPEG2000 sont en général maintenues. Par contre, lorsque la décomposition s'appuie sur un ré-échantillonnage de l'image, les performances objectives sont moins bonnes.

Dans un cadre de compression scalable, les performances des méthodes sont peu mises en avant. Certaines décompositions adaptatives produisent des sous-bandes qui peuvent être encodées par le codeur EBCOT de JPEG2000. Ceci permet de conserver les bonnes propriétés du codeur, dont la scalabilité spatiale et en qualité. D'une manière générale, la plupart des transformées étant multi-échelles, la scalabilité spatiale semble naturelle. Il serait cependant intéressant d'évaluer la pertinence visuelle des images de basses résolution spatiales générées. La scalabilité SNR quant à elle peut souvent être

mise en œuvre par un codage en plans de bits. Lorsque la géométrie est multi-échelles, elle peut aussi être encodée et décodée de manière scalable mais ceci nécessite de calculer des paramètres à chaque échelle.

Dans le chapitre 4, nous décrirons une nouvelle technique pour le codage adaptatif d'une image. A la manière des méthodes par déformation de blocs, cette technique s'appuie sur un ré-échantillonnage de l'image suivi par une décomposition classique horizontale-verticale. A la différence des méthodes par blocs, nous représentons la déformation par un maillage déformable. Ceci permet d'effectuer une déformation continue sur tout le domaine image et d'éviter ainsi un traitement particulier sur les bords des blocs.

Dans les travaux antérieurs, le maillage déformable a surtout été utilisé pour estimer un mouvement entre deux images. Avant d'exposer nos travaux sur l'image fixe, nous présentons dans le chapitre suivant les outils antérieurs qui permettent de modéliser le mouvement dans une vidéo et donc de s'adapter au contenu temporel. Ceci nous permet en particulier d'introduire le maillage déformable et les techniques d'estimation existantes. Nous montrons également qu'il existe différentes façons d'exploiter le mouvement dans un cadre de codage. Nous revoyons ainsi le codage prédictif et certains schémas par analyse-synthèse proposés dans le passé. Nos travaux des chapitres 4 et 5 s'inscrivent dans la continuité de ces schémas par analyse-synthèse.

Chapitre 3

Adaptivité temporelle dans les codeurs vidéo : outils antérieurs

Une vidéo naturelle est un signal 2D+t qui possède des corrélations dans l'espace mais aussi le temps. Au chapitre 2, nous avons décrit différents outils permettant de modifier le noyau d'analyse en fonction du contenu spatial d'une image fixe. Nous nous penchons maintenant sur les corrélations existant le long de l'axe temporel dans le cas d'une séquence vidéo. En effet, tout comme la géométrie définit des trajectoires de régularité spatiale dans le cas d'une image fixe, le mouvement apparent définit des trajectoires de régularité temporelle dans le cas d'une vidéo. Très tôt, l'exploitation de ce mouvement a paru intuitive et a été incluse dans les premières normes. Le principe est semblable au cas 2D : il s'agit d'orienter et d'allonger le noyau d'analyse temporel le long des lignes de flux optique. Dans ce chapitre, nous décrivons certains outils antérieurs permettant la modélisation, l'estimation et l'exploitation du mouvement dans une vidéo.

3.1 Modélisation paramétrique du champ de mouvement

3.1.1 Champ de mouvement unidirectionnel

Dans cette section, nous considérons le champ de mouvement défini par les variations de la fonction $I_t(\mathbf{x})$ entre deux instants : un instant courant t_c et un instant dit de référence t_r. Nous supposerons qu'il définit pour chaque pixel du domaine image $\mathcal{D}_{t_c}$ un vecteur déplacement $v^{t_c \to t_r}(\mathbf{x})$ donnant la direction de régularité maximale entre les deux instants [1]. A partir de ce champ de mouvement, il est possible de prédire l'image I_{t_c} à partir de I_{t_r}. L'image prédite est notée $\bar{I}_{t_c}$. Elle est donnée par :

$$\bar{I}_{t_c}(\mathbf{x}) = I_{t_r}(\mathbf{x} + v^{t_c \to t_r}(\mathbf{x})) \quad \forall \mathbf{x} \in \mathcal{D}_{t_c} \tag{3.1}$$

La prédiction de I_{t_c} à l'aide d'un champ de mouvement et d'une image de référence est aussi appelée *compensation en mouvement*. $\bar{I}_{t_c}$ est l'image compensée résultante.

[1] Notons qu'en toute rigueur le champ de mouvement réel peut rarement être défini partout sur le domaine image, notamment du fait des zones à occultation.

Dans la suite, nous allons présenter différents modèles pour représenter un champ de mouvement entre deux images. Cette modélisation doit satisfaire le même compromis entre adaptivité et parcimonie que la modélisation géométrique vue au chapitre 2. Ici, l'adaptivité est la capacité à représenter une large gamme de mouvements. Elle peut se mesurer en évaluant la prédiction de l'image I_{t_c} à l'aide d'un certain critère (en général, l'erreur quadratique ou la somme des différences absolues). Les modèles présentés ci-dessous associent un mouvement à des blocs de pixels. Ils se distinguent par le nombre de paramètres disponibles pour caractériser le mouvement et par les contraintes de régularité sur le domaine image.

Comme nous le verrons dans la section 3.3 et dans nos travaux exposés au chapitre 5, certains algorithmes s'intéressent aussi à la qualité de l'image reconstruite à l'instant de référence t_r en inversant la compensation en mouvement. Or, du fait des zones à occultation qui génèrent des discontinuités dans le champ de mouvement réel, le champ inverse ne peut être défini partout de façon correcte. Nous tâcherons donc également de différencier les modèles en fonction de la qualité de reconstruction qu'ils permettent.

3.1.2 Modèle translationnel par blocs

Considérons une partition du domaine image $\mathcal{D}_{t_c}$ en blocs de taille constante. Cette partition est notée $\mathcal{B}$. Le modèle de mouvement par blocs le plus largement utilisé (notamment dans les normes) associe à tous les pixels d'un bloc $b \in \mathcal{B}$ le même vecteur mouvement noté $v_b^{t_c \to t_r}$ (figure 3.1). Pour chaque pixel $\mathbf{x}$ dans un bloc $b \in \mathcal{B}$, l'image prédite $\bar{I}_{t_c}$ est alors donnée par :

$$\bar{I}_{t_c}(\mathbf{x}) = I_{t_r}(\mathbf{x} + v_b^{t_c \to t_r}) \tag{3.2}$$

Pour simplifier les notations, nous noterons dans la suite $v_b^{t_c \to t_r} = v_b$. En utilisant ce modèle, on suppose que chaque bloc de l'image est animé d'un mouvement de translation. L'adaptation au flux optique dépend donc de la taille des blocs : plus ils sont petits, plus l'hypothèse a des chances d'être validée. Dans les standards vidéo, une image est découpée en blocs de taille 16×16 dits *macro-blocs*. Pour permettre une meilleure adaptation aux disparités du champ de mouvement, des *modes* ont été introduits. Chaque mode consiste en un re-découpage particulier de chaque macro-bloc en blocs de taille variable et est choisi pour optimiser un critère débit-distorsion [Ric03].

Le modèle de mouvement par blocs bénéficie de plusieurs avantages. En particulier, puisque chaque bloc se déplace indépendamment de ses voisins, un tel modèle s'avère très efficace pour représenter les discontinuités de mouvement aux frontières des objets. D'autre part, un seul paramètre est nécessaire pour représenter le mouvement dans un bloc ce qui limite le surcoût de l'adaptivité. Enfin, le découpage en blocs permet un traitement des blocs en parallèle ce qui a abouti à des implémentations VLSI (« Very-Large-Scale Integration ») à faible complexité.

Un tel modèle présente aussi quelques limitations. Notamment, il s'avère inadapté dès qu'un objet de la scène est animé d'un mouvement de rotation ou de remise à l'échelle (dans le cas d'un zoom par exemple). En outre, le déplacement indépendant de blocs voisins peut générer des discontinuités à la frontière de ces blocs dans l'image

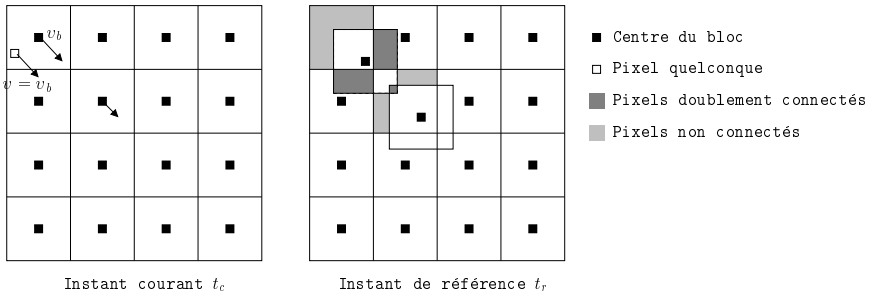


Fig. 3.1 : Modèle translationnel par blocs et zones problématiques lors d'une compensation en mouvement inverse.

prédite. Lors d'une approximation, ceci se traduit par un phénomène de blocs désagréable. Ce phénomène peut être réduit en utilisant un filtre de « deblocking » après compensation [Wie03, ZRMZ05]. Une autre solution consiste à utiliser le modèle par blocs recouvrants présenté dans le paragraphe suivant.

Le champ de mouvement défini par un modèle par blocs n'est pas réversible car pas bijectif. Plus précisément, si l'on souhaite reconstruire une intensité en chaque pixel de l'image I_{t_r} à partir de $\bar{I}_{t_c}$, on se heurte à deux problèmes. D'une part, certains pixels de $\mathcal{D}_{t_r}$ n'ont pas de correspondant dans $\mathcal{D}_{t_c}$. Ces pixels sont dits *non connectés* (figure 3.1). D'autre part, certains pixels de $\mathcal{D}_{t_r}$ ont plusieurs correspondants dans $\mathcal{D}_{t_c}$. Ils sont dits *multiplement connectés*. La présence de pixels non connectés s'explique essentiellement par l'apparition de régions à l'instant t_r qui sont occultées à l'instant t_c. De tels pixels ne peuvent donc être reconstruits par la seule donnée de $\bar{I}_{t_c}$. A contrario, la présence de pixels multiplement connectés s'explique par l'occultation de régions à l'instant t_r qui sont apparentes à l'instant t_c. Par définition, plusieurs valeurs sont candidates pour reconstruire l'intensité d'un tel pixel. Nous verrons en section 3.3 comment les auteurs gèrent ces cas.

3.1.3 Modèle translationnel par blocs recouvrants

Le modèle de mouvement par blocs recouvrants [OS94, SM00] noté OBMC pour « Overlapped Block Motion Compensation » a été proposé de manière à atténuer les phénomènes de blocs. Il est par exemple utilisé dans la norme H.263 [GFS97] et dans le codeur basé-ondelettes proposé par le groupe VidWav dans le cadre MPEG [AhG05]. Désormais, le domaine image $\mathcal{D}_{t_c}$ est découpé en blocs qui se recouvrent comme le montre la figure 3.2. Comme précédemment, à chaque bloc b de la partition est associé un et un seul vecteur mouvement v_b. L'élément distinctif par rapport au modèle précédent est que chaque pixel de l'image à prédire est maintenant connecté à plusieurs positions dans le domaine de référence.

Chaque pixel $\mathbf{x} \in \mathcal{D}_{t_c}$ peut en effet être associé à une liste de vecteurs mouvement v_{b_i} où $\{b_i\}$ est l'ensemble des blocs auxquels $\mathbf{x}$ appartient. Chacun de ces vecteurs mouvements donne une valeur de prédiction possible $\bar{I}_{t_c,i}(\mathbf{x}) = I_{t_r}(\mathbf{x} + v_{b_i})$. La valeur

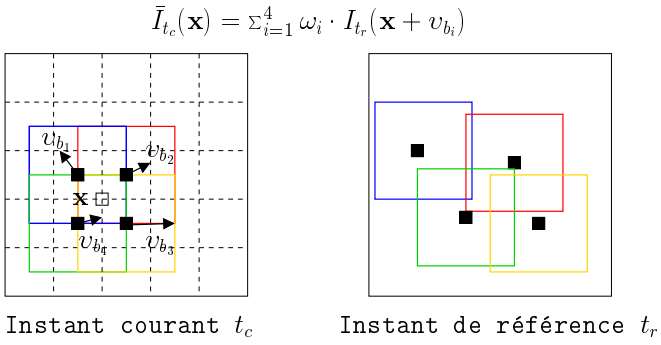


Fig. 3.2 : Modèle translationnel par blocs recouvrants. Chaque pixel de l'image à prédire est connecté à plusieurs positions dans le domaine de référence.

finale prédite est alors calculée en effectuant une combinaison linéaire de ces valeurs candidates. En associant un ensemble poids $\{\omega_i\}$ à l'ensemble des vecteurs $\{v_{b_i}\}$ vérifiant $\sum_i \omega_i = 1$, on peut écrire :

$$\bar{I}_{t_c}(\mathbf{x}) = \sum_i \omega_i \cdot \bar{I}_{t_c,i}(\mathbf{x}) = \sum_i \omega_i \cdot I_{t_r}(\mathbf{x} + v_{b_i}) \tag{3.3}$$

D'après la formule précédente, on notera que la compensation par blocs recouvrants n'a pas pour but de lisser le champ de mouvement. Elle opère un lissage des *intensités* qui permet d'atténuer les effets de blocs. Le résultat dépend de la taille des blocs recouvrants et des poids ω_i. En général, le poids associé au vecteur v_b d'un bloc b dépend de la distance du pixel au centre de ce bloc. Ce poids est déterminé par une fenêtre de lissage qui vaut 1 au centre du bloc et décroît en se déplaçant vers ses bords. Une des problématiques de l'OBMC est de trouver un compromis adéquat entre lissage du phénomène de blocs et conservation des discontinuités dans les zones à occultation notamment. Certains papiers comme [AKOK92, OS94] ont montré que la fenêtre bilinéaire offre les meilleures performances parmi différentes fenêtres fixes.

Dans le cas de l'OBMC, l'inversion de la compensation reste un problème ouvert. Observons simplement que le fait d'avoir des blocs recouvrants limite le nombre de pixels non connectés et augmente le nombre de pixels multiplement connectés. Des outils simples pour gérer ces zones seront utilisés au chapitre 5.

3.1.4 Blocs déformables

3.1.4.1 Déformations d'un bloc pour compensation

Dans les deux modèles précédents, chaque bloc b est animé d'un mouvement de translation caractérisé par un vecteur mouvement v_b. En notant (d_1, d_2) les composantes de ce vecteur, chaque pixel $\mathbf{x} = (x, y)$ d'un bloc dans $\mathcal{D}_{t_c}$ peut ainsi être mis en correspondance avec une position $\mathbf{x}' = (x', y')$ dans $\mathcal{D}_{t_r}$ selon la transformation spatiale à deux paramètres :

$$\begin{aligned} w : \mathcal{D}_{t_c} &\rightarrow \mathcal{D}_{t_r} \\ (x, y) &\mapsto (x', y') = (x + d_1, y + d_2) \end{aligned} \tag{3.4}$$

Et l'image prédite à l'intérieur du bloc dans $\mathcal{D}_{t_c}$ s'écrit :

$$\bar{I}_{t_c}(\mathbf{x}) = I_{t_r}(w(\mathbf{x})) \quad \forall \mathbf{x} \in b \tag{3.5}$$

D'autres modèles de transformation peuvent être définis pour englober une gamme plus large de mouvement. Par exemple, la transformation affine à 6 paramètres est donnée par :

$$w(x, y) = (a_1 x + a_2 y + d_1, a_3 x + a_4 y + d_2) \tag{3.6}$$

En plus des translations, cette transformation permet de modéliser le mouvement de rotation d'un bloc mais aussi la déformation d'un bloc (carré ou rectangle) en un parallélogramme.

Dans les travaux que nous avons menés, à la fois pour l'image fixe (chapitre 4) et pour la vidéo (chapitre 5), nous avons utilisé la déformation de blocs (facettes d'un maillage quadrangulaire dans notre cas) pour paramétrer des transformations bilinéaires. La transformation bilinéaire a 8 paramètres s'écrit :

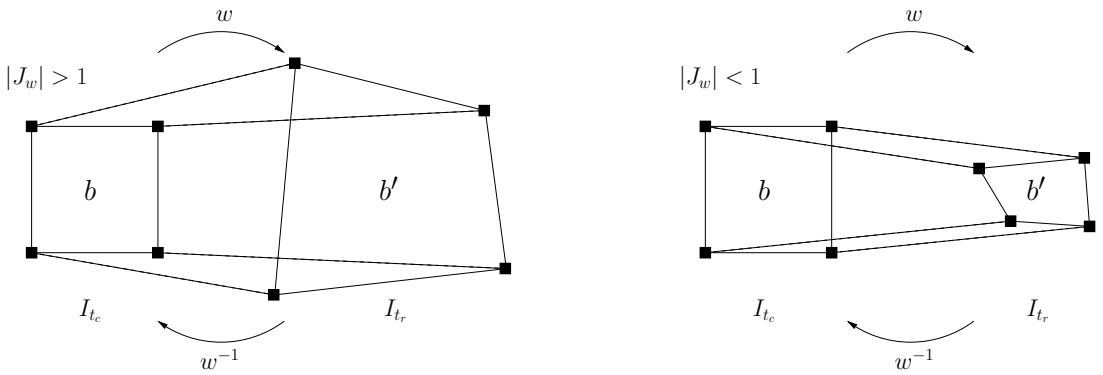
$$w(x, y) = (a_1 x + a_2 y + a_3 xy + a_4, a_5 x + a_6 y + a_7 xy + a_8) \tag{3.7}$$

Elle permet de modéliser des mouvements plus complexes que la translation, comme par exemple les changements d'échelle (zoom avant/arrière) dus au déplacement de la caméra ou les rotations. Elle est largement utilisée car elle fournit en général de bons résultats tout en limitant la complexité des calculs. Les 8 paramètres peuvent être déterminés par les positions des 4 sommets du bloc dans $\mathcal{D}_{t_r}$.

Nous voyons donc qu'il est possible de généraliser le modèle par blocs pour représenter des mouvements plus complexes que la translation. Néanmoins, multiplier par 3 ou 4 le nombre de paramètres à transmettre par bloc s'avère très coûteux. Pour limiter le nombre de paramètres, on peut imposer des contraintes de continuité aux frontières des blocs. Le maillage régulier et les modèles hybrides présentés ci-après forcent ainsi des sommets de blocs voisins à rester connectés lors de la transformation spatiale. Le degré de contrainte fixe le compromis entre adaptivité et parcimonie. Il détermine aussi la proportion de pixels non connectés et multiplement connectés apparaissant lors d'une compensation inverse. Notons enfin que les propriétés algébriques et géométriques des transformations introduites ci-dessus sont développées dans le livre de Wolberg [Wol94].

3.1.4.2 Gain et perte de résolution par rapport à t_r

Supposons que l'image à l'intérieur d'un bloc carré b dans $\mathcal{D}_{t_c}$ soit prédite avec les valeurs d'un bloc de forme quelconque b' dans $\mathcal{D}_{t_r}$, en utilisant par exemple l'une des transformations ci-dessus. Dans ce paragraphe, nous nous intéressons à la qualité de



(a) Perte de résolution (b) Gain de résolution

Fig. 3.3 : Perte et gain de résolution lors de la prédiction.

la compensation inverse obtenue en reconstruisant les valeurs du bloc b' avec la seule donnée des valeurs prédites à l'intérieur du bloc b. La qualité de la compensation inverse est notamment importante dans les schémas de codage par analyse-synthèse décrits au paragraphe 3.3.3.3 et dans nos travaux présentés dans les chapitres suivants. Cette qualité dépend principalement des changements de résolution lors du passage de $\mathcal{D}_{t_r}$ à $\mathcal{D}_{t_c}$.

Ces changements de résolution peuvent être mesurés localement à l'aide du jacobien J_w de la transformation w. La définition du jacobien a été donnée au chapitre précédent (paragraphe 2.3.4.1). Dans le cas de la transformation affine, le jacobien est constant sur l'ensemble du bloc et correspond au rapport des aires de b' et b. Dans le cas de la transformation bilinéaire, il dépend de la position à l'intérieur du bloc b. Soit $\mathbf{x}$ un point de $\mathcal{D}_{t_c}$. Les trois cas suivants peuvent se produire :

- Si $|J_w(\mathbf{x})| > 1$ alors il y a une *perte de résolution* lors du passage de $\mathcal{D}_{t_r}$ à $\mathcal{D}_{t_c}$. C'est la configuration de la figure 3.3(a) où un bloc b' de l'image à l'instant de référence est compensé à l'instant courant sur un bloc b contenant moins d'échantillons. On parlera aussi de *contraction* du bloc.
- Si $|J_w(\mathbf{x})| < 1$ alors il y a un *gain de résolution* lors du passage de $\mathcal{D}_{t_r}$ à $\mathcal{D}_{t_c}$. C'est la configuration de la figure 3.3(b) où un bloc b' de l'image à l'instant de référence est compensé à l'instant courant sur un bloc b contenant plus d'échantillons. On parlera d'*étirement* du bloc.
- Si $|J_w(\mathbf{x})| = 1$, *dans le cas général* il n'y a pas de changement de résolution lors du passage de $\mathcal{D}_{t_r}$ à $\mathcal{D}_{t_c}$. Par exemple, si w est une translation ou une rotation, alors son jacobien est unitaire. Notons que cette propriété ne garantit nullement la reconstruction parfaite de b'. En particulier, la rotation d'un bloc dans le domaine spatial se traduit également par une rotation dans le domaine fréquentiel. Comme le support fréquentiel d'un signal discret est compris dans un carré $[-\pi, \pi]^2$, une telle rotation implique forcément une perte (figure 3.4). Notons enfin qu'il existe un *cas particulier* où $|J_w(\mathbf{x})| = 1$ n'est pas synonyme de conservation de la résolution.

C'est le cas où la déformation w est donnée par $w(x, y) = (Kx, y/K)$ où K est une constante non nulle. Une telle déformation provoque un changement de résolution inverse dans les deux directions mais a un jacobien unitaire.

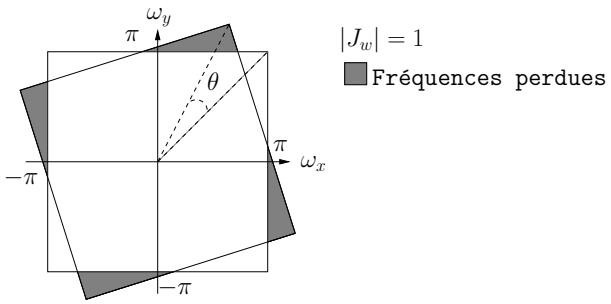


Fig. 3.4 : Pertes fréquentielles lors de la rotation d'un signal.

Si l'on souhaite reconstruire les valeurs de I_{t_r} sur b' avec la seule donnée des valeurs compensées sur b, alors les ré-échantillonnages effectués lors de la compensation et de la compensation inverse produisent nécessairement la perte de certaines fréquences. Cependant l'impact numérique et l'impact *visuel* de cette perte dépendent beaucoup de la valeur du jacobien et du contenu du bloc. Nous verrons pourquoi la prise en compte de ces pertes est un défi important des méthodes par analyse-synthèse.

3.1.5 Maillage déformable ou « Control Grid Interpolation » CGI

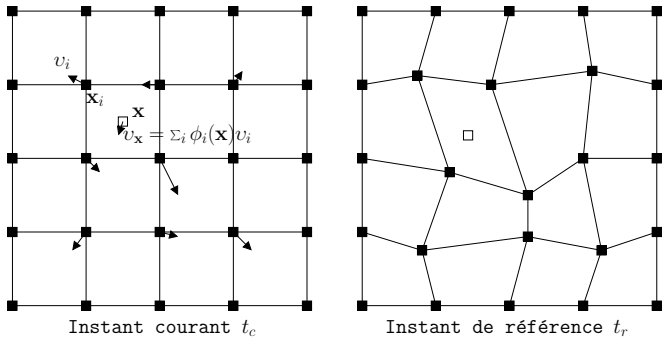


Fig. 3.5 : Modèle de mouvement par maillage déformable.

Au chapitre précédent, nous avons vu que le maillage pouvait servir de grille d'échantillonnage adaptative pour approximer une image avec des éléments finis. Dans le cas du mouvement, le maillage est souvent utilisé comme un *modèle déformable* : les positions de ses nœuds déterminent le mouvement associé à chaque pixel entre l'instant courant et l'instant de référence. On parle aussi de *grille de contrôle* (« Control Grid ») [SB91].

Dans ce paragraphe, nous supposons que toutes les facettes du maillage sont connectées et couvrent le domaine image (figure 3.5). Notons N_s le nombre de sommets du maillage. Les coordonnées $\mathbf{x}_i = (x_i, y_i)$ d'un nœud $i \in \{1 \dots N_s\}$ à l'instant courant sont fixées a priori et les paramètres du modèle sont les positions correspondantes notées $\mathbf{x'}_i = (x'_i, y'_i)$ à l'instant de référence. Le vecteur mouvement associé à un sommet i est quant à lui noté υ_i. Ses composantes sont données par la différence entre les positions du sommet aux deux instants, c'est-à-dire $\upsilon_i = (\Delta x_i, \Delta y_i) = \mathbf{x'}_i - \mathbf{x}_i$. Étant donnés les mouvements des nœuds du maillage, le mouvement d'un pixel quelconque $\mathbf{x}$ du domaine $\mathcal{D}_{t_c}$ peut être interpolé à l'aide d'une fonction de forme 2D ϕ :

$$\upsilon_{\mathbf{x}} = \sum_i \phi(\mathbf{x} - \mathbf{x}_i)\upsilon_i \tag{3.8}$$

Pour simplifier les notations, nous noterons dans la suite $\phi(\mathbf{x} - \mathbf{x}_i) = \phi_i(\mathbf{x})$. Les composantes d'un vecteur mouvement en un pixel $\mathbf{x}$ quelconque s'écrivent alors :

$$\begin{cases} \Delta x = \sum_i \phi_i(\mathbf{x}) \cdot \Delta x_i \\ \Delta y = \sum_i \phi_i(\mathbf{x}) \cdot \Delta y_i \end{cases} \tag{3.9}$$

et l'image prédite $\bar{I}_{t_c}$ est donnée par :

$$\bar{I}_{t_c}(\mathbf{x}) = I_{t_r}(\mathbf{x} + \sum_i \phi_i(\mathbf{x})\upsilon_i) \tag{3.10}$$

Cette expression peut être comparée à la prédiction donnée par l'OBMC (3.3). Ici, c'est le champ de mouvement qui est lissé par la fonction de forme, et non les niveaux de gris.

Les sommets du maillage sont reliés par des arêtes pour former des facettes, le plus souvent triangulaires ou quadrilatérales. Dans le cas du triangle, la fonction de forme utilisée en général est linéaire, elle vaut 1 au centre d'un triangle et décroît linéairement jusqu'à valoir 0 sur les arêtes. Ceci revient à modéliser la déformation à l'intérieur de la maille par la transformation affine donnée à l'équation (3.6). Avec ce type de transformation il est très simple de mettre en correspondance deux positions même si les triangles aux deux instants sont arbitraires. Dans le cas du quadrilatère, la fonction de forme bilinéaire est souvent choisie. Si un carré à l'instant t_c est mis en correspondance avec un quadrilatère quelconque à t_r, il est aisé de calculer le mouvement des pixels à l'intérieur de la facette. Cependant, lorsque les deux quadrilatères sont arbitraires, le calcul doit se faire en deux temps en passant par un carré ou un rectangle, ce qui requiert le calcul d'une fonction bilinéaire inverse non rationnelle. Ces considérations sont traitées dans l'article de Wang et Lee [WL96a].

Le maillage présente quelques avantages comparé aux modèles par blocs déconnectés. En particulier, la transformation affine ou bilinéaire permet de modéliser des mouvements plus complexes que la translation tout en conservant un nombre similaire de paramètres au total. La densité des nœuds peut aussi être adaptée au mouvement local en utilisant par exemple une structure hiérarchique de type Quadtree [LW95]. Dans ce

cas la connectivité du maillage n'est plus régulière et les décisions de subdiviser ou non les branches du Quadtree doivent être transmises en plus des positions des nœuds.

Lorsque les facettes du maillage restent connectées, elles définissent des correspondances bijectives entre l'instant courant et l'instant de référence sur l'ensemble du domaine image continu. De ce fait, il est possible d'inverser les correspondances sans se heurter au problème des pixels non connectés ou multiplement connectés. Dans le cas de mailles quadrangulaires, l'inversion des correspondances nécessite un peu plus de calculs que dans le cas de mailles triangulaires. Wang et Lee [WL96a] donnent les formules permettant d'inverser les correspondances entre un carré et un quadrilatère quelconque.

L'inconvénient principal d'avoir des mailles partout connectées est de ne pouvoir modéliser que des mouvements continus sur le domaine image. Ceci limite la qualité de la prédiction dans les zones à occultation où le mouvement réel est discontinu. Les mouvements de rotation ne peuvent pas non plus être modélisés efficacement. Pour répondre à ces limitations, l'idée est de casser la structure dans les zones mal prédites. C'est par exemple l'atout du modèle *SCGI* introduit ci-après.

3.1.6 Modèles hybrides SCGI et SOBMC

Pour mieux prendre en compte les discontinuités de mouvement, différentes méthodes ont été proposées. Une solution consiste à traiter les objets d'une scène séparément en leur attribuant chacun un maillage propre. Cette solution a par exemple été étudiée par Altunbasak [Alt97], Van Beek et Tekalp [BT97] ou encore Lechat [Lec99b]. Elle s'intègre bien dans la norme MPEG-4 permettant un codage objets [MPE02]. Une autre solution consiste à partir d'un maillage connecté sur tout le domaine image puis à casser la structure uniquement dans les zones à occultations. C'est le concept de *lignes de rupture* exploité par Marquant [Mar00] et Cammas [Cam04b]. Le long d'une ligne de rupture, le maillage est prolongé de part et d'autre et permet de modéliser des recouvrements ou découvrements de zones. Notons cependant que toutes ces méthodes ont recours à une segmentation avant ou pendant l'estimation de mouvement. Les modèles hybrides SCGI et SOBMC ne nécessite pas de segmentation.

Le modèle SCGI pour « Switched Control Grid Interpolation » est proposé par Ishwar et Moulin dans [IM00]. Le principe est illustré sur la figure 3.6. Désormais, le mouvement d'une maille de l'instant courant à l'instant de référence peut être modélisé soit par une translation soit par une déformation (bilinéaire par exemple). Un label est associé au nœud supérieur gauche de chaque maille pour dicter le choix du modèle. Considérons un nœud i. Dans la figure 3.6, un label 1 signifie que tous les nœuds de la maille restent connectés aux mailles incidentes lors de la transformation spatiale. Un label 0 signifie que les 3 nœuds différents de i se déconnectent du maillage pour suivre le même mouvement que i. Par rapport au maillage régulier, on voit donc qu'une nouvelle information de connectivité s'ajoute à la représentation. Cette information additionnelle est cependant limitée à 1 bit par nœud. Le modèle SCGI offre en outre une meilleure adaptivité au mouvement : il permet de modéliser des déformations continues (Label 1 : modèle CGI) tout en tolérant des cassures lorsque le mouvement réel est trop discontinu (Label 0 : modèle BM). Ces cassures apparaissent notamment dans les zones

occultées. Elles permettent une meilleure prédiction de l'image I_{t_c}. Bien sûr, le fait de casser la structure refait apparaître le problème des pixels non connectés ou multiplement connectés dans le cas d'une compensation inverse. Notons que dans [HMCP01], une méthode similaire est suivie par les auteurs mais le choix local du modèle se fait entre le CGI et l'OBMC.

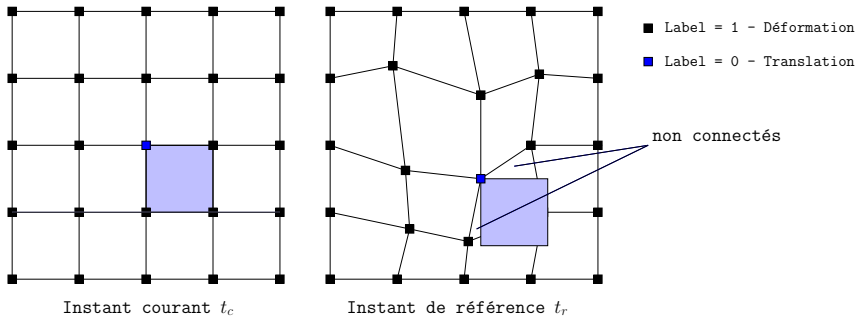


Fig. 3.6 : Principe du SCGI. Un label est associé à chaque bloc pour décider s'il est mieux prédit à l'aide d'une translation ou d'une déformation.

En suivant le principe de labellisation, Ishwar et Moulin [IM00] proposent également un modèle hybride entre le Block Matching et l'OBMC appelé « Switched OBMC » SOBMC. Ici, le label (0 ou 1) détermine si un pixel à l'intérieur d'un bloc est prédit à l'aide d'une seule valeur en suivant le vecteur mouvement du bloc uniquement (comme dans le cas du BM), ou de plusieurs valeurs en suivant les vecteurs mouvement des blocs voisins (comme dans le cas de l'OBMC). Les auteurs comparent l'efficacité des modèles hybrides SCGI et SOBMC par rapport au BM, à l'OBMC et au maillage régulier (CGI) dans une application de codage vidéo. Leurs résultats semblent suggérer que le SCGI apporte le meilleur compromis débit-distorsion sur une large gamme de débits et de formats d'image.

3.2 Estimation des paramètres de mouvement

Dans cette section, nous nous concentrons sur l'estimation des paramètres de mouvement lorsque le modèle choisi est l'un de ceux décrits précédemment. De très nombreuses approches existent pour estimer le mouvement. Le livre référence de Tekalp [Tek95] s'arrête sur les approches les plus communes : les méthodes basées sur l'équation du flux optique, celles basées blocs (block-matching et corrélation de phase), les méthodes dites « pel-récursives » basées sur le raffinement itératif des vecteurs déplacements ou encore les méthodes Baysiennes basées sur des modèles probabilistes. Ici, nous nous penchons sur un nombre limité de techniques que nous avons implémentées dans le cadre des travaux présentés au chapitre 5.

3.2.1 Block Matching

L'algorithme de « Block Matching » est l'outil d'estimation le plus utilisé en pratique du fait de sa facilité d'implémentation logicielle et matérielle. Considérons une partition en blocs du domaine image à l'instant courant. L'estimation du déplacement d'un bloc b est guidée principalement par deux critères [Tek95] :

Le critère de mise en correspondance. Les techniques de Block Matching comparent deux blocs pixel à pixel. Plusieurs critères peuvent être choisis : corrélation croisée (covariance), erreur quadratique moyenne (EQM), différence absolue moyenne (MAD), nombre de pixels correspondants (« matching pel count »)...

L'EQM et la MAD sont les deux critères les plus utilisés dans la littérature et correspondent respectivement à l'hypothèse d'une distribution Gaussienne ou Laplacienne a l'intérieur du bloc. En utilisant la MAD comme critère, le but de l'algorithme est de trouver le vecteur mouvement $v_b^\star$ qui satisfait :

$$v_b^\star = \arg\min_{v_b} \sum_{\mathbf{x}\in b} |I_{t_c}(\mathbf{x}) - I_{t_r}(\mathbf{x} + v_b)| \tag{3.11}$$

La stratégie de recherche. Plusieurs stratégies peuvent être appliquées pour rechercher le vecteur $v_b^\star$ dans une fenêtre de recherche pré-définie. La technique la plus basique est de faire une recherche exhaustive. Si cette technique permet de trouver le vecteur optimal, elle est aussi très gourmande en temps (et calculs) en particulier si la recherche est faite avec une précision sous-pixellique. Il est souvent nécessaire de trouver un juste compromis entre complexité et précision de l'estimation. C'est pourquoi des stratégies de recherche itératives ont été proposées. Initialement le vecteur nul est choisi comme optimal. L'idée est de tester un même nombre (limité) de candidats autour du vecteur optimal courant à chaque itération mais en réduisant progressivement la fenêtre de recherche (figure 3.7). C'est le principe des procédures de recherche dites « n-step search » [KIH+81, LZL94], « log-D search » [JJ81], « cross search » [Gha90] ou « diamond search » [ZM00]. Une recherche itérative par descente en gradient peut également être mise en place [LF96].

Lorsque l'instant courant et l'instant de référence sont très éloignés, il est nécessaire d'agrandir la fenêtre de recherche. Dans ce cas, pour accélérer l'estimation une technique hiérarchique s'appuyant sur la pyramide multi-résolutions des images peut être utilisée. La résolution la plus petite permet d'estimer de grands déplacements (relativement à la résolution d'origine). Ces déplacements servent à initialiser l'estimation au niveau de résolution supérieur et ainsi de suite jusqu'à obtenir une précision pixellique ou sous-pixellique à la résolution de l'image. On parle de *technique multi-résolutions*.

Notons enfin qu'une technique d'estimation hiérarchique peut également être mise en place pour limiter les discontinuités de mouvement entre des blocs voisins. Cette technique revient à estimer le mouvement de blocs dyadiques à chaque niveau d'une partition en Quadtree. L'estimation commence à un niveau du Quadtree sur des blocs de grande taille. Au niveau suivant, le mouvement de chaque bloc est initialisé avec

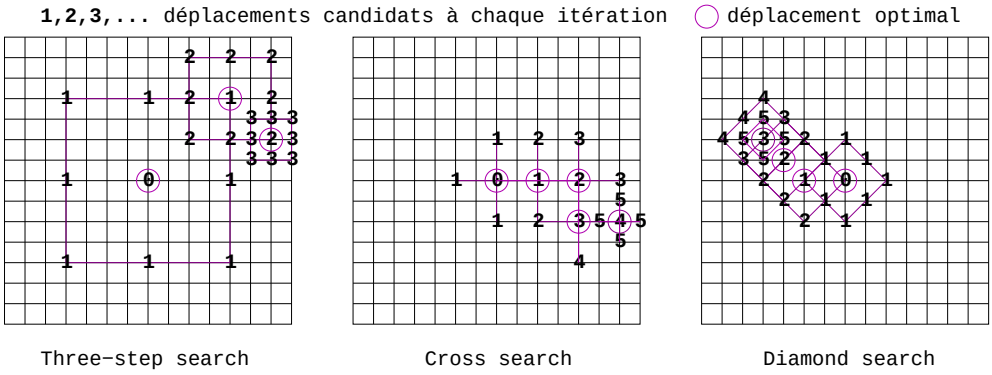


Fig. 3.7 : Trois stratégies de recherche du déplacement optimal.

le mouvement du bloc parent puis est raffiné en poursuivant l'estimation. On parle de *technique multi-grilles*. En suivant une approche lagrangienne, similaire à celle décrite au chapitre précédent paragraphe 2.3.2, il est également possible d'adapter localement la taille du bloc en fonction du compromis débit-distorsion désiré.

3.2.2 OBME

Si l'on souhaite effectuer une compensation par blocs recouvrants, les vecteurs estimés lors d'un Block Matching ne correspondent plus à l'erreur de prédiction réellement observée. C'est pourquoi des techniques d'estimation spécifiques au modèle OBMC existent. Elles peuvent comme les techniques précédentes intégrer un compromis entre erreur de prédiction et coût des vecteurs. Un éventail de ces algorithmes est donné par Su et Mersereau [SM00]. Le problème principal d'une telle estimation est que le déplacement d'un bloc modifie l'erreur de prédiction dans tous les blocs voisins. De ce fait une dépendance non-causale existe entre les vecteurs mouvement voisins, ce qui signifie qu'aucun algorithme tournant en temps polynomial ne peut estimer les vecteurs optimaux de tous les blocs [CW96].

L'algorithme de Block Matching fenêtré WBMA [NO92] est une simple extension du BMA. Le WBMA estime le mouvement d'un bloc recouvrant b indépendamment de ses voisins. Si on écrit ϕ la fonction de forme définie sur le bloc et qui vaut 1 en son centre, le WBMA cherche à réduire l'erreur de prédiction sur b définie par :

$$\mathbf{E}(v_b) = \sum_{\mathbf{x}\in b} \{\phi(\mathbf{x})[I_{t_c}(\mathbf{x}) - I_{t_r}(\mathbf{x}+v)]\}^2 \tag{3.12}$$

D'autres algorithmes proposent d'estimer le mouvement des blocs selon un ordre de parcours bien défini, par exemple de type raster [SM97] ou en damier [KCK96]. Le mouvement du bloc couramment traité est alors calculé en prenant en compte les mouvements des blocs voisins déjà connus.

Enfin, des algorithmes itératifs existent : après une initialisation des vecteurs (par exemple en utilisant un BMA ou WBMA), le principe est de segmenter le champ de

vecteurs en différents groupes disjoints. Pour chaque groupe, une estimation OBME est effectuée en fixant les autres groupes. Le procédé se répète jusqu'à convergence (ou jusqu'à ce qu'un nombre maximal d'itérations soit atteint). Par exemple, l'algorithme *ICM* (« Iterative Conditional Mode ») [OS94, AKOK92] propose de diviser le champ de mouvement en quatre groupes définis de sorte que, étant donnés trois groupes, les blocs dans le quatrième groupe ne se recouvrent pas. L'algorithme s'appuie sur la théorie des champ de Markov. Il est adapté à des implémentations parallèles et converge rapidement, mais il est aussi gourmand en calculs, très sensible à l'initialisation des vecteurs et aboutit fréquemment à un minimum local. L'algorithme *IRow* (« Iterative Row Optimization ») [CW96] propose une alternative en optimisant les vecteurs par rangée. Il s'appuie sur une programmation dynamique et réduit la probabilité de rester bloqué en un minimum local.

3.2.3 Maillage régulier

3.2.3.1 Estimation type matching

Dans les algorithmes de Block Matching précédents, le vecteur mouvement d'un bloc est recherché parmi plusieurs candidats possibles qui sont testés tour à tour par rapport au critère choisi. Le même type de méthode peut être mis en place pour estimer le mouvement avec un maillage régulier. On rappelle que dans le cas du maillage régulier, les paramètres à estimer sont les positions des nœuds dans le domaine référence $\mathcal{D}_{t_r}$. Tous les blocs (ou mailles) sont inter-connectés et animés d'un mouvement de déformation. Si un nœud se déplace dans $\mathcal{D}_{t_r}$, il modifie la prédiction de plusieurs mailles dans $\mathcal{D}_{t_c}$. Le domaine d'influence d'un nœud dépend du support de la fonction de forme introduite à l'équation (3.8). Du fait de la dépendance entre mailles, une recherche exhaustive voudrait que l'on considère toutes les combinaisons possibles de déplacements des nœuds, ce qui n'est pas raisonnable en pratique.

Dans [SB91], Sullivan et Baker proposent une alternative. Le principe est de déplacer chaque nœud l'un après l'autre - en fixant tous les autres - de façon à minimiser un critère de correspondance sur le domaine d'influence du nœud courant uniquement. Pour ce faire, on teste de façon exhaustive tous les déplacements possibles du nœud dans une fenêtre de recherche donnée. Puisque le déplacement d'un nœud peut modifier la position optimale de ses voisins, la méthode proposée est itérative. Un flag noté `local_opt` est associé à chaque nœud et spécifie si l'optimum local est atteint. A chaque itération, uniquement les nœuds dont le tag est `false` sont considérés. Si le déplacement optimal courant trouvé est identique à celui de l'itération précédente, le tag est mis à `true`. Sinon, le tag est mis à `false`, ainsi que les tags de tous les nœuds dans le voisinage $\mathcal{N}$ du nœud courant. L'algorithme s'arrête lorsque tous les tags sont `true` ou qu'un nombre maximal d'itérations a été atteint (voir l'algorithme 1 donné ci-dessous).

Dans l'approche proposée par Sullivan et Baker, les facettes du maillage sont quadrangulaires. Une technique similaire a été proposée par Nakaya et al. [NH91] pour un maillage triangulaire. Elle est appelée « Hexagonal Search » car le voisinage $\mathcal{N}$ du nœud courant comprend maintenant 6 nœuds. La seule différence notable est que le dépla-

cement des nœuds est initialisé avec une technique de Block Matching. L'Hexagonal Search a ensuite été largement repris, par exemple par Altunbasak et al. [AT97a] qui y ont ajouté une technique de relaxation multi-grilles pour accélérer les calculs.

Notons enfin que des techniques d'estimation hiérarchique s'appuyant sur la mise en correspondance ont également été développées. Citons en particulier les travaux de Huang et Hsu [HH94] ou de Hsiang et al. [HMCP01] qui permettent d'estimer un maillage Quadtree.

Algorithme 1 Raffinement itératif des vecteurs mouvements v_i, $i \in \{1 \dots N_s\}$

```
local_opt(i) ← false   ∀i ∈ {1...N_s}
v_i = 0   ∀i ∈ {1...N_s}
k ← 0 // Itération
repeat
  k ← k + 1
  compteur ← 0
  for i = 1 à N_s do  // Pour chaque nœud
    compteur ← compteur + 1
    if local_opt(i) == false then
      compteur ← compteur + 1
      v_i^old ← v_i
      v_i ← v_i^⋆ // Trouver v_i optimal dans la fenêtre de recherche
      if v_i^old == v_i then
        local_opt(i) ← true
      else
        local_opt(j) ← false   ∀j ∈ 𝒩(i)
      end if
    end if
  end for
until compteur == 0 ou k ⩾ k_max
```

3.2.3.2 Minimisation énergétique

Les méthodes précédentes trouvent les paramètres optimaux en testant un ensemble de paramètres candidats. Dans le cas du maillage, des techniques de résolution *analytiques* existent. Elles consistent à exprimer le critère de correspondance comme une fonction des paramètres puis à chercher le minimum de la fonction à l'aide d'un outil d'optimisation mathématique, par exemple la descente en gradient. Wang et Lee [WL94, WL96a, WL96b] ont détaillé de nombreuses problématiques liées à l'estimation de mouvement par maillage. En particulier, dans [WL94], ils formulent le critère **E** à minimiser comme une somme pondérée de différentes énergies, chacune liée à une problématique spécifique. Dans le cadre de nos travaux, deux énergies ont été utilisées :

$\mathbf{E}_d$ est une énergie interne au maillage dite énergie de déformation. Elle est introduite pour éviter que le maillage ne se déforme trop pendant l'estimation par rapport au maillage uniforme initial. La notion de déformation est exprimée mathématiquement en considérant chaque arête du maillage comme un ressort ayant une certaine constante. En affectant la même constante à tous les ressorts, le maillage est vu comme un système

physique dont l'état d'équilibre est l'état initial. Plus le poids associé à cette énergie est fort, plus les nœuds sont contraints par le mouvement de leurs voisins. Il faut noter que dans ce cas, les positions des nœuds du maillage sont plus régulières et présentent donc un coût de codage moins élevé. En contrepartie, la minimisation de $\mathbf{E}$ met en général plus de temps à converger. Les auteurs proposent d'adapter la constante des ressorts en fonction du contenu local de façon à restreindre moins le déplacement des nœuds dans les régions de contours où le mouvement réel est en général moins régulier.

$\mathbf{E}_m$, appelée erreur de « matching » par les auteurs est l'erreur entre l'image courante I_{t_c} et la prédiction $\bar{I}_{t_c}$. Wang et Lee utilisent l'erreur quadratique :

$$\mathbf{E}_m = \sum_{\mathbf{x}} [I_{t_c}(\mathbf{x}) - \bar{I}_{t_c}(\mathbf{x})]^2 \tag{3.13}$$

En remplaçant les prédictions $\bar{I}_{t_c}(\mathbf{x})$ par leur expression (3.10), on peut exprimer cette erreur en fonction des paramètres $\upsilon_i \quad \forall i \in \{1 \dots N_s\}$ du modèle :

$$\mathbf{E}_m = \sum_{\mathbf{x}} [I_{t_c}(\mathbf{x}) - I_{t_r}(\mathbf{x} + \sum_i \phi_i(\mathbf{x})\upsilon_i)]^2 \tag{3.14}$$

Les deux autres énergies prises en compte par les auteurs sont notées $\mathbf{E}_f$ et $\mathbf{E}_i$. $\mathbf{E}_f$ est utilisée si l'on souhaite suivre une caractéristique particulière d'une image à l'autre. Les auteurs s'intéressent par exemple au suivi des contours. $\mathbf{E}_i$ est l'erreur d'approximation commise en interpolant les valeurs de l'image à l'intérieur des facettes uniquement avec les valeurs aux nœuds. En effet, dans leurs travaux, les auteurs considèrent le maillage déformable non seulement comme modèle de mouvement mais aussi comme grille d'échantillonnage à chaque instant (voir chapitre 2, paragraphe 2.3.4).

Les paramètres du modèle minimisant la somme de toutes ces énergies peuvent être déterminés *de façon globale* en effectuant par exemple une descente en gradient. La dérivation de l'énergie total $\mathbf{E}$ par rapport aux paramètres, puis l'annulation de la dérivée, donne un système linéaire de type $A \cdot \mathbf{X} = B$ où A est une matrice creuse de dimensions $2N_s \times 2N_s$. Nous montrons en annexe B comment construire un tel système en dérivant $\mathbf{E}_m$. Le nombre de coefficients dans chaque ligne de A ne dépend que du support de la fonction de forme. Par exemple, si la fonction de forme est la fonction bilinéaire, alors chaque ligne ne contient que 9 coefficients non nuls car le déplacement d'un nœud n'influence que le déplacement de ses voisins d'ordre 1. Pour de telles matrices, des méthodes de résolution de système linéaire très performantes existent. Par exemple les méthodes par gradient conjugué dont la complexité dépend du nombre de nœuds N_s de façon quasi-linéaire.

Notons que les poids associés à chaque énergie sont des paramètres de l'algorithme. Leurs valeurs peuvent être déterminées pratiquement en appliquant l'estimateur sur des vidéos tests. Notons également que l'estimation peut être raffinée comme dans les méthodes précédentes en utilisant une méthode hiérarchique pour adapter la taille des mailles au mouvement local [WL96b]. Notons enfin qu'il est possible d'estimer le mouvement pour optimiser la prédiction de l'image courante (problème d'analyse) tout en assurant une bonne reconstruction de l'image de référence par compensation inverse

(problème de synthèse). Nous renvoyons par exemple le lecteur aux travaux de Marquant et al. [MPL00c] qui ont proposé une méthode nommée « backward in forward » car elle intègre le compromis analyse-synthèse.

3.2.3.3 Problème de conformité

Au cours de l'estimation, une maille peut devenir dégénérée. Nous avons vu au paragraphe 2.3.4.1 du chapitre précédent qu'une maille est dégénérée si le jacobien de la déformation en un point par rapport à l'élément maître est inférieur ou égal à 0. Dans le cas d'un suivi de mouvement, si les mailles sont triangulaires, ceci se produit dès que la position 2D d'un nœud sort de l'hexagone formé par ses quatre voisins incidents, provoquant alors le retournement d'une maille. Si les mailles sont quadrangulaires, il suffit qu'un nœud sorte du quadrilatère formé par ses voisins. Lorsqu'un maillage cesse d'être conforme, le mouvement modélisé n'est plus bijectif ce qui pose problème lorsque l'on souhaite effectuer une compensation en mouvement inverse. Par ailleurs, les cas de dégénérescences peuvent rendre les algorithmes instables. L'énergie de déformation $\mathbf{E}_d$ permet de limiter leur apparition mais elle ne les empêche pas. Plusieurs techniques existent pour contraindre la conformité du maillage, voir par exemple la thèse de Lechat [Lec99b] (chapitre 16).

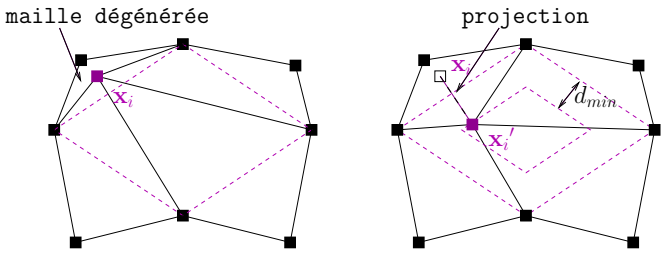


Fig. 3.8 : Projection non-obtuse. D'après [WL94].

Dans les travaux décrits aux chapitres 4 et 5, nous avons eu recours à la méthode proposée dans [WL94] appelée « projection non-obtuse ». Cette technique observe chaque sommet i et utilise le quadrilatère formé par ses quatre voisins incidents pour détecter une dégénérescence : si la position $\mathbf{x}_i$ du sommet se trouve à l'extérieur du quadrilatère, cela signifie qu'une maille a un sommet obtus et est dégénérée. Pour y remédier, les auteurs proposent donc de replacer le sommet dans le quadrilatère par une projection orthogonale sur le côté le plus proche. Comme un nœud placé exactement sur le quadrilatère aura tendance à en sortir, les auteurs préconisent une projection sur un quadrilatère plus petit (voir figure 3.8).

3.2.4 Modèles hybrides

Dans [IM00], Ishwar et Moulin proposent des méthodes simples pour calculer les paramètres des modèles SCGI et SOBM. Pour le SCGI, l'idée est d'étendre l'algorithme 1.

Désormais, pour chaque nœud et chaque vecteur mouvement candidat, il s'agit de tester aussi les 2 labels de connection possibles. Remarquons que la valeur d'un label ne modifie la prédiction que dans un seul bloc incident au nœud dans $\mathcal{D}_{t_c}$, noté b : si le label est 1 le bloc est prédit par un warping, si le label est 0 il est prédit par une translation. A chaque itération la combinaison (label,vecteur) donnant la plus petite erreur de prédiction est retenue. D'une itération à l'autre, il est possible que le label optimal soit modifié car le déplacement des autres nœuds de b auront été mis à jour. Les auteurs précisent que l'algorithme est gourmand en calculs. Cependant, il est garanti de converger en un nombre fini d'itérations car il existe un nombre fini de combinaisons de labels et de vecteurs mouvement candidats et que l'erreur de prédiction décroît au sens large à chaque itération et a une borne inférieure (0). Pour le modèle SOBM, la technique peut être encore plus directe. Il s'agit de calculer indépendamment les paramètres d'un modèle par blocs et les paramètres d'un modèle par blocs recouvrants. Ensuite, le choix du label pour chaque nœud est effectué en comparant l'erreur de prédiction obtenue dans les deux cas.

3.3 Exploitation du mouvement dans les codeurs

Dans les sections précédentes, nous avons présenté différents *modèles* de mouvement et avons donné des outils pour *estimer* les paramètres de ces modèles. Nous nous intéressons maintenant à la façon dont peut être *exploité* ce mouvement pour le codage d'une séquence vidéo.

3.3.1 Codage prédictif basique

Le codage prédictif est à la base de tous les standards H.26x proposés par l'ITU-T et MPEG-x proposés par l'ISO-IEC. Une séquence vidéo est découpée en groupe d'images ou GOF (« Group Of Frames ») de taille N_G. Les premières images de chaque GOF sont des images clés appelées images *Intra*. Ces images sont encodées indépendamment des autres images de la séquence sans prédiction temporelle. Les images Intra permettent de maintenir un certain niveau de qualité dans la séquence décodée et offrent un point d'ancrage dans une vidéo. Toutes les autres images d'un GOF sont appelées images *Inter* et sont prédites en exploitant le mouvement. Un schéma de base avec boucle de prédiction est présenté figure 3.9.

Les images Inter notées P sont prédites à partir de l'image qui les précède dans le GOF, soit une image Intra I ou une autre image P. Si I_t est l'image courante à prédire, alors I_{t-1} est l'image de référence. La prédiction $\bar{I}_t$ est obtenue en effectuant une estimation de mouvement entre t et $t-1$. Le résidu de prédiction $I_t - \bar{I}_t$ est ensuite calculé. Il est encodé puis transmis avec les paramètres de mouvement. Au décodage, l'image I_t est décodée après l'image I_{t-1}. A l'aide du mouvement décodé, on peut retrouver l'image $\bar{I}_t$ prédite à l'encodage et donc reconstruire I_t en lui ajoutant le résidu de prédiction décodé.

Les images Inter *bidirectionnelles* notées B sont prédites à partir de deux images, I ou P, en estimant un mouvement *bidirectionnel*. Plus précisément, les blocs d'une

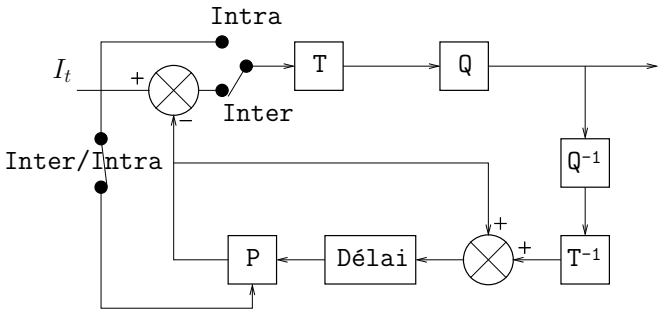


Fig. 3.9 : Schéma de principe d'un codeur avec boucle de prédiction. T : Transformée 2D (DCT, ondelettes...). Q : Quantification. P : Prédiction temporelle.

image courante peuvent être prédits à partir de deux images de référence, passées ou futures. Ceci requiert de modéliser et d'estimer un champ de mouvement entre l'image courante et chacune des deux images prises comme référence. Ces deux images doivent être encodées et décodées avant l'image courante pour garantir sa reconstruction, ce qui nécessite une organisation particulière des données.

Dans le dernier standard H.264/AVC [H.203], le concept d'images P et B a été généralisé. Désormais, une P peut être prédite à partir d'une image quelconque dans une liste d'images préalablement encodées, I, P ou B, passées ou futures. De la même façon, une image B est prédite à partir de deux images quelconques dans une liste d'images préalablement encodées, I, P ou B, passées ou futures. Ceci améliore les performances en termes de compression mais est moins robuste aux erreurs de reconstruction.

Les standards de codage vidéo sont des codeurs hybrides où les images Intra et les résidus de prédiction sont transformés avec une DCT par blocs et où le modèle de mouvement utilisé est le modèle par blocs (translationnel). Depuis les premiers standards H.261 [SG193] et MPEG-1 [MPE93, Sik97], le schéma avec boucle de prédiction n'a cessé d'être optimisé. Les améliorations concernent toutes les briques de transformée et de codage, l'organisation des images I,P,B ainsi que l'ajout de fonctionnalités comme la « scalabilité ». Un état de l'art des standards avec les optimisation successives est donné dans la thèse de Robert [Rob08]. Avec le dernier standard H.264/AVC qui résulte d'un effort conjoint entre l'ITU-T et l'ISO-IEC, les performances en compression ont encore été multipliées par 2 par rapport au codeur MPEG-2.

3.3.2 Codage hybride basé ondelettes 3D

Parallèlement à l'optimisation du schéma prédictif des standards, un intérêt croissant a été porté aux techniques de décomposition basées ondelettes 3D. L'idée est d'exploiter les corrélations le long d'une trajectoire de mouvement sur *plusieurs* images à l'aide d'une ondelette temporelle ; puis de décomposer chaque image de sous-bande temporelle avec une ondelette 2D. On parle de codage « t+2D ». La décomposition temporelle est souvent désignée par l'acronyme MCTF pour « Motion Compensated Temporal Filtering ». Outre les bonnes propriétés de décorrélation de l'ondelette, le caractère

multi-résolutions d'une représentation en ondelettes motive fortement les travaux des chercheurs car elle semble offrir une réponse naturelle à la scalabilité.

3.3.2.1 Codage QMF et transformée de Haar

Dans cette section, on suppose qu'une estimation de mouvement est effectuée entre chaque image d'un GOF et l'image qui la précède prise comme référence. Le mouvement estimé pour chaque image est uni-directionnel. L'application d'une ondelette le long d'une trajectoire de mouvement pose deux problèmes principaux. Tout d'abord, si le modèle de mouvement utilisé n'est pas bijectif alors un pixel non connecté à l'instant de référence n'appartient à aucune trajectoire de mouvement future, tandis qu'un pixel multiplement connecté à l'instant de référence appartient à plusieurs trajectoires de mouvement futures. La question est de savoir comment prendre en compte ces pixels pour garantir leur reconstruction au décodage. Ensuite, si l'on souhaite estimer des vecteurs mouvement avec une précision sous-pixellique, le problème se complexifie car l'intensité en un pixel de l'instant courant peut être prédite par interpolation de différentes valeurs à l'instant de référence : tout se passe comme si plusieurs pixels à l'instant de référence appartenaient à la même trajectoire de mouvement future. Des solutions aux deux problématiques précédentes doivent être trouvées car d'une part, un modèle de champ non bijectif représente mieux les disparités de mouvement dans une scène naturelle et d'autre part, il est connu qu'un mouvement sous-pixellique donne de meilleures performances en termes de compromis débit-distorsion [CHRW03].

Avant que le schéma lifting n'offre des solutions à ces problématiques (voir plus bas), Ohm [Ohm94] puis Choi et Woods [CW99] ont proposé des approches s'appuyant sur le schéma de décomposition ondelette avec des filtres miroirs en quadrature de phase (QMF). Les deux approches peuvent être appliquées avec des filtres ondelettes de support quelconque. Nous nous limitons ici à l'ondelette de Haar car elles deviennent vite très complexes lorsque le support du filtre s'étend, en particulier du fait des problèmes cités plus haut. La figure 3.10 comparent les deux approches proposées lorsque le mouvement a une précision pixellique.

Ohm calcule l'image basse fréquence à l'instant courant t_c et l'image de résidus à l'instant de référence t_r. Lorsqu'un pixel à l'instant de référence est connecté à plusieurs pixels à l'instant courant, l'auteur choisit un correspondant pour calculer la haute fréquence, en général le premier rencontré dans l'ordre de parcours. Les autres correspondants à t_c deviennent isolés. Pour ces pixels devenus isolés, l'auteur propose de conserver la valeur originale comme valeur de basse-fréquence pour assurer leur reconstruction au décodage. Pour un pixel non connecté, il choisit de ne pas prendre un correspondant à l'instant courant car un tel pixel appartient généralement à une région de l'image qui disparaît. Au lieu de cela, il propose de choisir le correspondant à l'instant $t_r - 1$ en utilisant le mouvement préalablement estimé entre t_r et $t_r - 1$. Comme I_{t_r-1} est encodée et décodée avant I_{t_r}, le procédé peut être inversé au décodage.

Choi et Woods quant à eux proposent de calculer l'image basse fréquence à l'instant de référence et l'image de résidus à l'instant courant. Ceci permet de limiter la présence de pixels non connectés lors du calcul de l'image de résidus et ainsi réduire l'énergie

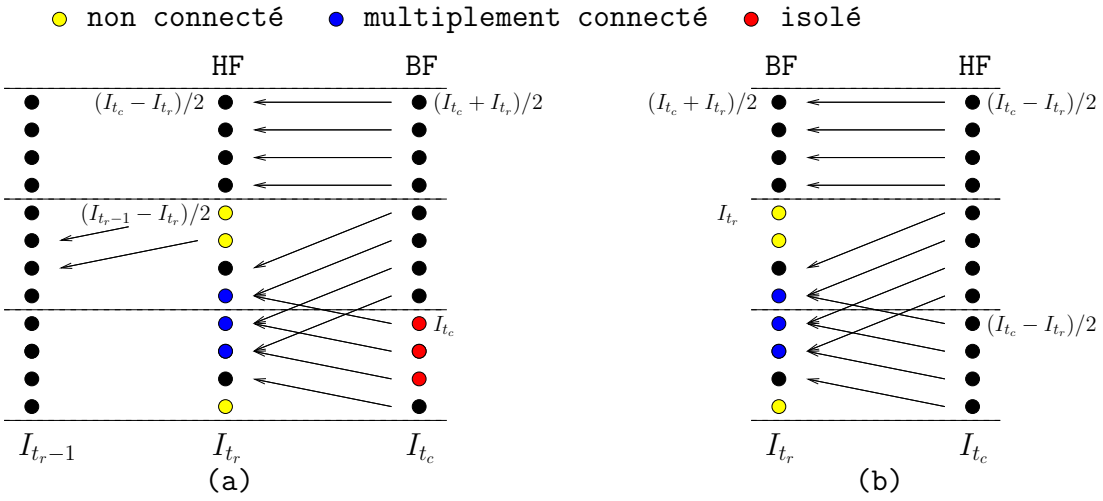


Fig. 3.10 : Transformée temporelle avec mouvement par blocs : (a) Schéma de Ohm [Ohm94], (b) Schéma de Choi et Woods [CW99].

des hautes fréquences. Lors du calcul de la basse fréquence, la valeur originale de I_{t_r} est conservée pour les pixels non connectés. Le cas des pixels multiplement connectés est résolu comme précédemment. Par contre, il n'est pas nécessaire ici d'isoler les correspondants d'un pixel non choisis : le résidu en ces pixels est calculé comme pour les autres pixels.

Remarquons que les approches présentées précédemment n'assurent par la reconstruction parfaite dans le cas de mouvements à précision sous-pixellique.

3.3.2.2 Threads de mouvement et lifting

En 2001, Xu et al. [XXLZ01] introduisent la notion de « thread » de mouvement. Une thread de mouvement peut être définie comme une ligne de flux temporelle qui débute et s'achève selon le mouvement des objets. Le principe est illustré sur la figure 3.11. Ici, un mouvement est estimé entre chaque image d'un GOF et l'image qui lui *succède*. Le modèle de mouvement utilisé est un modèle de mouvement par blocs et la précision est *pixellique*. Les vecteurs mouvement calculés permettent de lier plusieurs pixels le long d'une même ligne de flux. Une décomposition multi-résolutions avec l'ondelette 9/7 ou 5/3 le long des lignes de flux est alors proposée. Le problème des pixels non connectés ou multiplement connectés est bien sûr toujours présent. Si plusieurs pixels sont liés au même pixel dans l'image suivante, seulement l'un d'entre eux est rattaché à la ligne de flux. Les autres sont marqués comme pixels terminaux (« terminating pixels ») et les lignes de flux auxquelles ils étaient liés s'achèvent. Si certains pixels dans une image de référence ne sont connectés à aucun pixel dans l'image précédente, alors ils sont marqués comme pixels isolés (« non-referred pixels ») et indiquent le commencement d'une nouvelle ligne de flux.

Les auteurs mettent en avant deux limites du schéma. Tout d'abord, le schéma

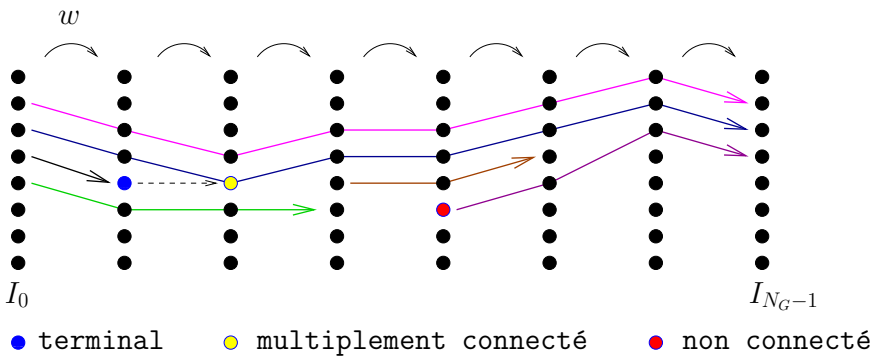


Fig. 3.11 : Exemples de threads de mouvement sur un GOF.

impose l'indépendance des lignes de flux : deux lignes de flux ne peuvent fusionner ou se diviser. Lorsque le mouvement est complexe, le nombre de pixels multi-connectés ou non connectés augmente et avec eux le nombre de lignes de flux qui s'achèvent ou débutent. Or, chaque terminaison ou début d'une ligne de flux introduit des effets de bord qui réduisent les performances en compression [XXLZ00]. Ensuite, la technique précédente utilise une précision pixellique. Or comme nous l'avons dit plus haut, il est connu qu'un mouvement sous-pixellique donne de meilleures performances en termes de compromis débit-distorsion.

L'introduction du schéma lifting [PPB01] a fourni une solution à un certain nombre de problématiques. Le principe est similaire à celui du lifting directionnel vu au chapitre 2 paragraphe 2.2.3 pour l'exploitation de la géométrie dans une image. La différence est que le filtrage est appliqué le long d'une ligne de flux temporelle et non plus géométrique. La division polyphase de la vidéo sépare les images paires des images impaires. Le lifting consiste dans un premier temps à prédire les images impaires à l'aide des deux images paires voisines pour calculer une sous-bande temporelle haute-fréquence. Dans un deuxième temps, les images paires sont mises à jour avec les images de détails aux deux instants impairs voisins calculés à l'étape précédente. Selon le support de l'ondelette choisie, les étapes de prédiction et de mise à jour peuvent être itérées. Pour mener à bien le schéma lifting directionnel, Secker et Taubman [ST01] proposent de calculer un champ de mouvement bi-directionnel (c'est-à-dire un mouvement entre t_c et $t_c + 1$ et entre t_c et $t_c - 1$) pour chaque image de la séquence. Ceci permet d'éviter l'apparition de pixels non connectés ou multiplement connectés dans l'image à prédire ou à mettre à jour. Cependant, ceci accroît aussi le coût de codage de l'information annexe. Pesquet-Popescu et Bottreau [PPB01] s'inscrivent dans la suite du schéma de Choi et Woods avec ondelette de Haar décrit plus haut et s'intéressent particulièrement au cas des pixels multiplement connectés intervenant lors du calcul de la basse fréquence (mise à jour d'une image impaire). Au lieu de choisir le premier correspondant dans l'ordre de parcours, ils proposent de sélectionner de manière adaptative la valeur correspondante la plus proche. Un critère de décision robuste à la quantification est mis en place pour retrouver le correspondant au décodage sans transmettre d'information supplémentaire.

En 2003, Luo et al. [LWLZ03] proposent un schéma en lifting directionnel pour répondre aux limitations des threads de mouvement tout en conservant la propriété de reconstruction parfaite. Dans ce schéma, une estimation de mouvement est réalisée entre chaque image impaire et les deux images paires qui l'entourent. Aucun mouvement partant d'une image paire n'est modélisé, de sorte que le nombre de modèles reste identique au schéma de Choi et Woods [CW99] et Xu et al. [XXLZ01]. Après avoir estimé le mouvement, à chaque étape de lifting *tout* pixel d'une image impaire ou paire peut désormais être rattaché à une trajectoire qui a débuté avant l'instant courant et se poursuit après. Un pixel précédemment marqué comme terminal peut être mis à jour en utilisant à la fois un correspondant à gauche et à droite. Ceci revient à « fusionner » deux lignes de flux au niveau du pixel multiplement connecté de l'image I_2 dans la configuration de la figure 3.11. Lors de la mise à jour de ce pixel, un seul correspondant est cependant utilisé dans l'image I_1 : le premier dans l'ordre de parcours du champ de mouvement. Ensuite, un pixel précédemment marqué comme isolé peut désormais être mis à jour en utilisant des correspondances dans les deux images impaires voisines. S'il n'est pas connecté à l'image précédente et/ou suivante, sa trajectoire de mouvement est choisie identique à une trajectoire voisine. La nature du schéma lifting garantit la reconstruction parfaite, et ce même si le mouvement estimé a une précision sous-pixellique.

Pour finir, remarquons que la décomposition temporelle en ondelettes par lifting compensé n'est pas forcément équivalente à la décomposition par bancs de filtres compensés. Dans sa thèse, André [And07] montre qu'il y a équivalence si les champs de mouvement sont inversibles et additifs. La condition d'additivité est rarement prise en compte et la condition d'inversibilité n'est pas satisfaite dans le cas d'un mouvement par blocs.

3.3.2.3 Barbell lifting

Dans le schéma de lifting directionnel précédent, nous avons mentionné que lorsqu'un pixel d'une image paire est connecté à plusieurs positions dans une image impaire, sa mise à jour ne prend en compte qu'un seul de ses correspondants. Pour améliorer la mise à jour, il est souhaitable de prendre en compte la contribution de tous les correspondants en accordant un certain poids à chacun d'entre eux. De plus, si l'on souhaite utiliser des modèles de mouvement par blocs recouvrants, un pixel d'une image paire se trouve aussi connecté à plusieurs positions dans les images impaires adjacentes. Pour prendre en compte ces nouvelles considérations, Xiong et al. [XWX+04, XXWL07] ont introduit la *fonction Barbell* et modifié le schéma de lifting précédent pour créer le *Barbell lifting* aussi appelé « Energy Distributed Update » (EDU).

Supposons connu le champ de mouvement entre une image impaire I_{2t+1} et l'image paire voisine I_{2t}. Pour tout pixel $\mathbf{a} \in \mathcal{D}_{2t+1}$, les auteurs définissent $\mathcal{M}_{2t+1\to 2t}(\mathbf{a}) \subset \mathcal{D}_{2t}$ comme l'ensemble des pixels dans $\mathcal{D}_{2t}$ auxquels $\mathbf{a}$ est connecté. Un poids $\omega(\mathbf{a},\mathbf{b})$ est associé à chaque paire de pixels $(\mathbf{a},\mathbf{b})$ ($\mathbf{b} \in \mathcal{M}_{2t+1\to 2t}(\mathbf{a})$). La valeur qui va servir à la prédiction de I_{2t+1} à gauche est alors donnée par la fonction barbell :

$$f_p^{2t+1\to 2t}(\mathbf{a}) = \sum_{\mathbf{b}\in\mathcal{M}_{2t+1\to 2t}(\mathbf{a})} \omega(\mathbf{a},\mathbf{b})\cdot I_{2t} \tag{3.15}$$

Les poids doivent vérifier la contrainte $\sum_{\mathbf{b}\in\mathcal{M}_{2t+1\to 2t}(\mathbf{a})}\omega(\mathbf{a},\mathbf{b}) = 1$ pour assurer la conservation de l'énergie. L'équation précédente permet de calculer une prédiction lorsqu'un pixel est multiplement connecté (cas de blocs recouvrants par exemple) mais englobe aussi le cas de la prédiction sous-pixellique. Les pixels connectés sont alors ceux à partir desquels la valeur prédite est interpolée. Le résidu de prédiction en $\mathbf{a}$ est donné par $d(\mathbf{a}) = I(\mathbf{a}) + \alpha_0 f_p^{2t+1\to 2t}(\mathbf{a}) + \alpha_1 f_p^{2t+1\to 2t+2}(\mathbf{a})$ où α_0 et α_1 sont les paramètres de prédiction dans l'étape courante du lifting.

L'étape de mise à jour est le pendant de l'étape de prédiction. Supposons qu'un pixel $\mathbf{a}\in\mathcal{D}_{2t+1}$ ait été associé à un pixel $\mathbf{b}\in\mathcal{D}_{2t}$ lors de la prédiction, avec un poids $\omega(\mathbf{a},\mathbf{b}) > 0$. Alors l'erreur de prédiction en $\mathbf{a}$ est utilisée pour mettre à jour la valeur en $\mathbf{b}$ avec le même poids. Pour tout pixel $\mathbf{b}\in\mathcal{D}_{2t}$, les auteurs définissent $\mathcal{M}_{2t\to 2t+1}(\mathbf{b}) = \{\mathbf{a}\in\mathcal{D}_{2t+1} | \mathbf{b}\in\mathcal{M}_{2t+1\to 2t}(\mathbf{a})\}$ comme l'ensemble des pixels à l'instant $2t+1$ auxquels $\mathbf{b}$ est connecté. La valeur qui va servir à la mise à jour du pixel $\mathbf{b}$ à droite est ainsi :

$$f_u^{2t+1\to 2t}(\mathbf{p}) = \sum_{\mathbf{a}\in\mathcal{M}_{2t\to 2t+1}} \omega(\mathbf{a},\mathbf{b})\cdot f_p(\mathbf{a}) \tag{3.16}$$

Et la valeur mise à jour en $\mathbf{a}$ est donnée par $a(\mathbf{a}) = I(\mathbf{a}) + \beta_0 f_p^{2t+1\to 2t}(\mathbf{a}) + \beta_1 f_p^{2t+1\to 2t+2}(\mathbf{a})$ où β_0 et β_1 sont les paramètres de mise à jour dans l'étape courante du lifting.

Pour terminer sur les approches par lifting directionnel, remarquons que beaucoup d'auteurs [LLL+01, VGP02, Pau06, And07] ont observé de meilleurs résultats de compression lorsque l'étape de mise à jour est désactivée. En particulier, l'ondelette 5/3 tronquée (c'est-à-dire sans mise à jour) aboutit à de meilleures performances que l'ondelette 5/3 ou l'ondelette 9/7. Elle est donc souvent choisie en pratique [AhG05]. Notons que la décomposition hiérarchique (appelée « Hierarchical B-Frames ») mise en place dans le dernier standard de codage « scalable » H.264/MPEG-4 SVC [JVT06] correspond elle aussi à une décomposition avec l'ondelette 5/3 tronquée.

3.3.3 Codage par analyse-synthèse

Les méthodes décrites précédemment s'appuient sur des boucles de prédiction voire de mise à jour dans le cas d'une transformée ondelette. Les coefficients de hautes et basses fréquences sont déterminés *en utilisant* les champs de mouvement estimés, et ce à chaque niveau de la transformée. Ces méthodes peuvent être associées aux méthodes vues pour l'image fixe qui cherchent à adapter le noyau de représentation à la géométrie de l'image. Ici, le but est d'adapter le noyau au mouvement existant dans la direction temporelle. Nous pouvons faire les mêmes remarques qu'au chapitre précédent : pour inverser la décomposition, le décodeur doit être parfaitement synchronisé avec le codeur, et en particulier le mouvement doit être connu précisément. Si le mouvement est décodé

avec perte alors les coefficients reconstruits après un niveau de décomposition inverse seront corrompus. Et sur J niveaux de décomposition les pertes s'accumulent. Pour mettre en place une scalabilité en mouvement, il est donc souvent nécessaire de modéliser un champ de mouvement pour chaque niveau de résolution temporelle. Notons toutefois que des travaux récents [ST04, AAAB06] proposent de prendre en compte les distorsions introduites par une perte sur le mouvement afin d'optimiser le partage du débit entre les coefficients d'ondelettes et les vecteurs mouvement. Ceci leur permet de générer un flux totalement scalable offrant des gains dans les bas débits.

Un des atouts des méthodes par analyse-synthèse présentées dans ce paragraphe est de casser la dépendance entre le mouvement et la décomposition temporelle. Par différenciation avec l'*analyse* harmonique, le terme *analyse* fait ici référence à un *pré-traitement* par exemple sur un GOF qui a pour but de construire un nouveau signal ayant un moindre coût de codage. La *synthèse* est le *post-traitement* qui permet de reconstruire une version du signal original après encodage et décodage du signal déformé.

3.3.3.1 Scène statique : modélisation 3D

Lorsqu'une scène est statique, le mouvement apparent dans la séquence vidéo est dû uniquement au déplacement de la caméra dans le temps. Supposons qu'un point 3D dans la scène filmée se projette en une position $\mathbf{x}_0 \in \mathcal{D}_{t_0}$ à un instant t_0 et en $\mathbf{x}_1 \in \mathcal{D}_{t_1}$ à un instant t_1. Connaissant le déplacement du point $\mathbf{x}_0$ entre t_0 et t_1, ainsi que les paramètres de la caméra, il est possible de reconstruire le point 3D en s'appuyant sur la *contrainte épipolaire* (voir [Bal05], chapitre 2). Partant de cette observation, Galpin [Gal02] propose d'exploiter le mouvement pour générer un modèle 3D valable pour chaque portion de la séquence. Le procédé est entièrement automatique. Il suppose que la scène est statique (ou segmentée au sens du mouvement), filmée par une caméra monoculaire dont les paramètres d'acquisition sont inconnus, que la distance focale est constante et que la scène ne contient pas de surface spéculaire. Les mouvements de la caméra sont supposés quelconques mais non dégénérés, c'est à dire qu'ils fournissent une information 3D (pas de rotation pure autour du centre optique par exemple).

La séquence est divisée en GOF. Deux GOF successifs partagent une image dite image *clé*. Les paramètres extrinsèques (position dans l'espace, rotation, translation) de la caméra pour les images clés sont estimés par une méthode de calibration classique couplée à un ajustement par faisceaux glissant adapté à la représentation (voir [GM02]). Les paramètres de la caméra entre les instants clés sont calculés par estimation de pose à l'aide de l'algorithme de Dementhon et Davis [DD95]. Le modèle 3D pour le GOF est construit à l'aide des correspondances entre les deux instants clés et des paramètres extrinsèques de la caméra à ces deux instants. Le modèle 3D est ensuite encodé et transmis. La première image du GOF qui constitue la *texture* du modèle est également encodée et transmise, de même que la position de la caméra à chaque instant. Au décodage, le modèle et sa texture sont décodés, puis la scène 3D est *synthétisée* en « plaquant » la texture sur le modèle. La portion de séquence peut alors être reconstruite en projetant la scène 3D sur chaque position de caméra décodée. Le principe général du décodage est donné figure 3.12.

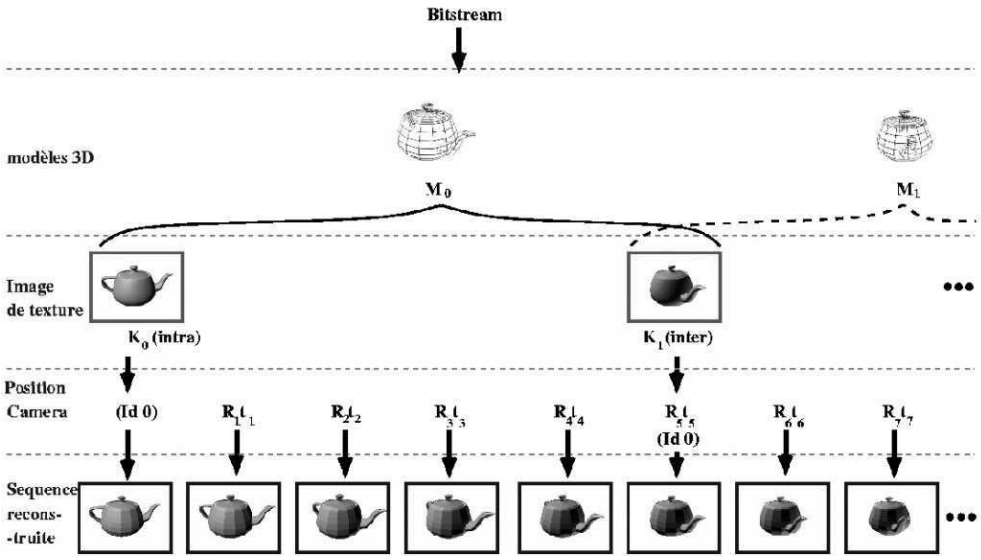


Fig. 3.12 : Reconstruction d'une séquence vidéo après modélisation 3D et transmission.

Pour reconstruire une séquence de bonne qualité, il n'est pas nécessaire de transmettre les modèles 3D sans perte. Les travaux de Balter [Bal05] exploitent les propriétés d'une telle représentation en termes de scalabilité. En codant le maillage 3D à l'aide d'ondelette géométrique (voir chapitre 2, paragraphe 2.3.4.3), l'auteur génère un flux emboîté décodé selon les capacités du réseau. A très bas débit, pour des scènes statiques, cette démarche offre une meilleure qualité visuelle que celle obtenue avec le codeur de l'état de l'art H.264 [BGM06]. Notons que le PSNR ne permet pas ici une bonne évaluation de la qualité car la méthode ne se focalise pas sur la reconstruction d'une image pixel à pixel : la séquence est reconstruite en effectuant un *rendu* d'une scène 3D à différents instants. Seules les images clés peuvent être reconstruites parfaitement. Pour assurer une transition douce lors du passage d'un GOF à un autre dans la séquence introduite, des outils spécifiques ont été introduits [GM02, BM03, GBMA04].

Remarquons que si la représentation par modèles 3D apporte de bonnes performances de compression, elle offre en outre de nouvelles fonctionnalités grâce au passage par la 3D : changements d'illumination, stabilisation de la séquence, insertion d'objets ou encore navigation libre (changement de trajectoire de la caméra par rapport au chemin original).

3.3.3.2 Technique mosaïques

Le codage vidéo basé *mosaïques* (on dit aussi basé *sprites*) est particulièrement efficace lorsque la caméra est affectée de mouvements panoramiques et/ou de zooms devant un arrière plan statique. Dans cette méthode, l'analyse permet de construire une image panoramique pour une portion donnée de la séquence. Dans un premier temps, chaque image est projetée (ou compensée en mouvement) dans un même système de coordon-

nées, typiquement le système de coordonnées de la première image. On fera souvent référence à l'instant de projection dans la suite, il sera noté t_p. Tour à tour, un mouvement est estimé entre I_{t_p} et chaque image du GOF prise comme image de référence. Les images projetées sont les différentes prédictions de I_{t_p} obtenues. Dans un second temps, l'analyse ré-échantillonne les images compensées et alignées puis les combine pour former une large image appelée *mosaïque*. La mosaïque est alors transformée avec une technique 2D (ondelettes par exemple). Elle est encodée puis transmise avec les mouvements estimés. A la synthèse, chaque image de la séquence peut être reconstruite à partir de la mosaïque en inversant le mouvement. Pour que cette inversion soit possible, le choix du modèle de mouvement se porte sur le maillage régulier qui permet une mise en correspondance globale et bijective.

Notons que ce type de méthodes n'est pas limité au cas des scènes statiques. En particulier, il est possible de segmenter le contenu d'une vidéo pour différencier l'arrière plan des objets en mouvement. On peut alors créer une mosaïque pour l'arrière plan et transmettre de façon indépendante les informations sur les objets. Il faut aussi transmettre un masque pour déterminer la position des objets à la synthèse. Nous pouvons renvoyer le lecteur par exemple aux travaux de Wang et Adelson [WA94], Pateux et al. [PMCM01] et Lee et al. [LCL+02]. Le codage objet basé sprite est également inclus dans le standard MPEG-4 [WJ01].

3.3.3.3 Compensation de mouvement global

La démarche utilisée précédemment peut être modifiée pour représenter une séquence vidéo animée sans passer par une segmentation objets. Dans [TZ94a], Taubman et Zakhor proposent ainsi de compenser toutes les images d'un GOF à un instant de projection. Du fait de la présence de régions occultées, il n'est pas possible de construire une seule mosaïque sans perdre de l'information. Cependant, si l'estimation de mouvement est efficace, les régions non occultées sont « alignées » dans le groupe d'images compensées. On peut dire que le but de la compensation est d'extraire le mouvement de façon explicite pour générer un GOF « sans » mouvement. Le GOF compensé est donc adapté à une décomposition « en ligne » le long de l'axe temporel. Cette approche est à relier à la technique par déformations de blocs proposée par les mêmes auteurs pour adapter les contours des images fixes à un filtrage horizontal/vertical (voir chapitre 2, paragraphe 2.2.4.2). Ici le but recherché est similaire : plutôt que d'adapter le noyau de décomposition aux trajectoire de mouvement, on cherche à adapter les trajectoires de mouvement à un filtrage standard. Le GOF compensé est encodé et transmis indépendamment du mouvement estimé. Après décodage, une version *synthétisée* des images d'origine peut être reconstruite en inversant la compensation opérée à l'analyse.

Le modèle de mouvement choisi par Taubman et Zakhor est un modèle *translationnel* global du plan image. Notons que théoriquement ce modèle permet d'inverser la compensation en mouvement sans perte en utilisant des filtres à réponses impulsionnelles infinies. Cependant, un tel modèle limite fortement la qualité des compensations. Dans [WXCM99], Wang et al. décrivent un schéma similaire mais cette fois-ci basé sur un modèle de mouvement par maillage. Comme nous l'avons vu au paragraphe 3.1.4.2,

lorsque le contenu d'une maille à un instant courant est prédit par une déformation autre que la translation, la prédiction peut provoquer des pertes ou des gains de résolution. Lors d'une compensation inverse, les pertes de résolution provoquent des chutes de qualité dans l'image reconstruite. Pour prendre en compte ce phénomène, Wang et al. proposent donc de définir l'instant de projection en fonction du mouvement de manière à limiter les pertes de résolution. L'objectif est que la résolution d'un pixel dans $\mathcal{D}_{t_p}$ corresponde à une résolution sous-pixellique dans le domaine des autres images du GOF. La figure 3.13 montre le cas d'une scène filmée par une caméra. La caméra est animée d'un mouvement de translation vers la droite, d'un mouvement de rotation dans le sens horaire et effectue un zoom avant puis arrière. L'instant de projection est indiqué en gris et correspond à l'instant où la scène est acquise à la résolution spatiale la plus élevée. En pratique, du fait de la présence de zones à occultation, il n'est pas possible de s'affranchir du problème des pertes de résolution si l'on raisonne à échantillonnage critique, et ce même si la scène est statique.

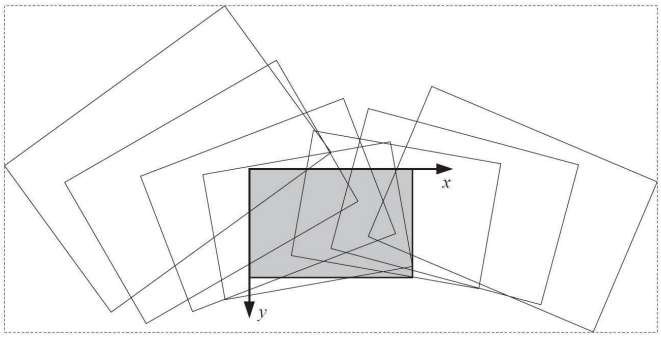


Fig. 3.13 : Groupe d'images projetées dans un système de coordonnées commun. D'après [WXCM99].

Dans les deux schémas présentés ci-dessus, un groupe d'images compensées est encodé après avoir subi une décomposition ondelettes 3D, c'est à dire que chaque image est décorrélée spatialement avant d'appliquer le filtrage temporel. Dans [Cam04b], Cammas s'intéresse à un schéma similaire mais optimisé pour une décomposition t+2D. Le modèle de mouvement est un modèle par maillage global. Différents modes d'analyse sont étudiés permettant d'utiliser une seule, deux, ou N_G grilles de projection. Dans ce dernier cas, le schéma revient à un schéma prédictif de type lifting. L'auteur montre que le nombre de compensations en mouvement nécessaires pour une décomposition en ondelettes liftées lorsqu'on utilise une seule ou deux grilles de projection est nettement moins important que dans le modèle prédictif. Un schéma de codage scalable est décrit dans lequel deux GOF successifs partagent une même image clé. Les deux images clés encadrant un GOF sont utilisées comme grilles de projection comme illustré figure 3.14. Pour générer un flux scalable temporellement, l'auteur met en place un encodage en plusieurs couches de résolution. Les deux images clés constituent la *couche de base* du GOF. Elles sont encodées avec JPEG2000 et placées en début de flux. Les autres images compensées sont tout d'abord prédites en effectuant une interpolation linéaire

des images clés (voir figure 3.15). Les résidus de prédiction sont ensuite décomposés à l'aide d'un filtrage ondelette aligné sur l'axe temporel. Le niveau de décomposition d'ondelettes J détermine le nombre de *couches de raffinement*. Les sous-bandes temporelles générées sont envoyées à JPEG2000 qui génère un flux scalable spatialement et en qualité. Notons que la prédiction linéaire effectuée avant la décomposition permet d'assurer une transition douce lors du passage d'un GOF à un autre après décodage et synthèse.

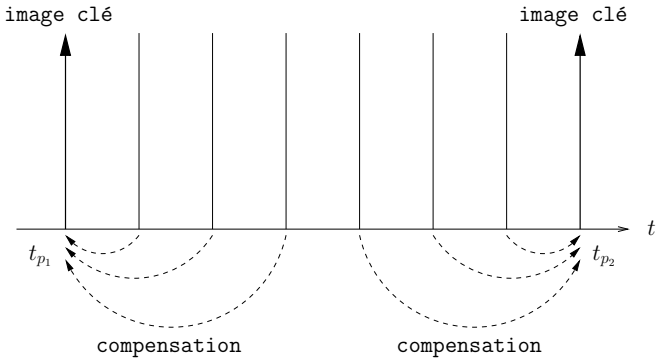


Fig. 3.14 : Projection des images d'un GOF sur deux grilles de référence. D'après [Cam04b].

Avant de conclure ce paragraphe, notons que l'évaluation de la qualité des séquences reconstruites par le PSNR n'est pas pertinente dans le cas des techniques ci-dessus. En effet ces techniques s'appuient sur des ré-échantillonnages successifs des images de départ et dans ces conditions l'objectif recherché ne peut être la reconstruction pixel à pixel. Il s'agit d'évaluer les gains et/ou pertes *visuels* générés par de telles approches.

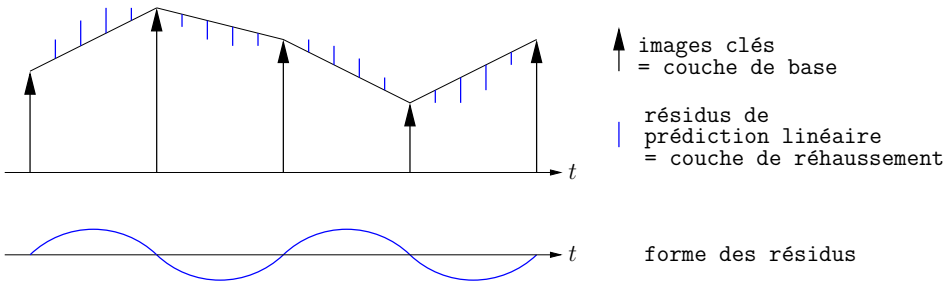


Fig. 3.15 : Structure en couches proposée par Cammas [Cam04b]. La couche de base comprend les images clés. Les images intermédiaires sont prédites par les images clés. Les résidus de prédiction ont une forme particulière qui permet une décomposition quasi-orthogonale même à la frontière entre GOF.

3.3.4 Remarques sur la scalabilité

Bien que déjà présente dans les standards de codage vidéo MPEG-2 [MPE94] et MPEG-4 [MPE02], la scalabilité n'était cependant pas assez efficace. Dans le dernier standard H.264/MPEG-AVC [H.203], l'introduction des images de type B hiérarchique permet une scalabilité temporelle performante. Une couche de base assurant un niveau de qualité minimal est constituée par les images intra I et quelques images P. Les couches de rehaussement temporel sont calculées en ajoutant des images B ou P entre chaque image et ainsi de suite jusqu'au dernier niveau de rehaussement. La décomposition et l'organisation des couches suit donc un processus « bottom-up » (voir figure 3.16(b)).

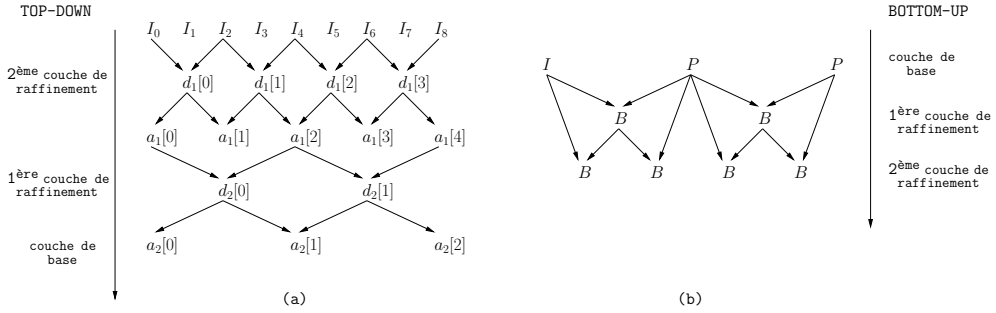


Fig. 3.16 : Schémas de décomposition temporelle type ondelette (a) et type MPEG (b).

En janvier 2005, les groupes MPEG et Video Coding Experts Group (VCEG) se sont entendus pour finaliser le projet SVC (« Scalable Video Coding ») comme un amendement du standard H.264/MPEG-4 AVC. Le nouveau standard H.264/MPEG-4 SVC [JVT06, SMW07], finalisé en juillet 2007, reprend le principe des images B hiérarchiques[2] mais intègre également la scalabilité spatiale et SNR. La structure en couches est toujours de type « bottom-up ». Une couche de base est construite en décimant la vidéo dans l'espace et le temps. Elle est encodée avec H.264/MPEG-4 AVC pour assurer une qualité minimale de basse résolution spatiale et temporelle. A partir de la couche de base, il est possible de prédire un niveau de résolution spatiale plus fin en sur-échantillonnant les images du niveau inférieur. Le résidu de prédiction donne une couche de rehaussement spatial. De même une couche de rehaussement en qualité SNR peut être calculée à partir de la couche de base pour apporter une scalabilité SNR. Différentes couches de rehaussement peuvent ainsi être ajoutées au flux binaire pour une même résolution temporelle. La résolution temporelle supérieure est atteinte en codant une couche de réhaussement comme dans le cas de H.264/MPEG-4 AVC. Notons qu'un inconvénient d'une structure de type « bottom-up » est que les différentes couches sont optimisées une par une du bas vers le haut. Ainsi, l'échec d'une prédiction inter-couche à un niveau peut réduire les performances aux niveaux plus fins.

Face à la structure en couche « bottom-up » des standards, la décomposition en

[2]En réalité les images B hiérarchiques ont été introduites spécifiquement pour le codeur SVC puis intégrées à AVC après avoir observé qu'elles amélioraient les performances du codeur non scalable.

ondelettes, comme celle utilisée dans le barbell lifting [XXWL07], offre une structure naturelle de type « top-down ». Cette structure est illustrée figure 3.16(a) pour la dimension temporelle. La vidéo est décomposée dans les 3 directions temporelle, horizontale et verticale pour générer des sous-bandes spatio-temporelles. Comme la transformée en ondelettes est à échantillonnage critique, le nombre de coefficients est le même que le nombre de pixels et les différents niveaux de résolution sont emboîtés les uns dans les autres. Cette représentation a l'avantage d'être multi-résolution et donc d'offrir une scalabilité naturelle. Toutefois, il est important de préciser qu'un codage après décomposition ondelettes t+2D offre des performances moindres par rapport au codeur SVC en termes de scalabilité spatiale. En effet, dans ces méthodes, la scalabilité spatiale est obtenue en supprimant des sous-bandes de haute fréquence spatiale dans chaque sous-bande temporelle. Or, ceci n'est pas équivalent à supprimer des sous-bandes de haute fréquence spatiale dans les images d'origine et aboutit à des artefacts gênants, particulièrement dans les zones où le mouvement est discontinu. Pour pallier à ce phénomène, Mehrseresht and Taubman [MT06] ont ainsi proposé une technique dite 2D+t+2D qui permet de ré-intégrer des hautes fréquences temporelles dans les sous-bandes spatiales et d'améliorer les résultats.

Scalabilité en mouvement. A moins de prendre en compte l'impact d'une perte de mouvement à chaque niveau de décomposition [ST04, AAAB06], la boucle de prédiction dans les codeurs standards et basés lifting ne permet pas de décoder un mouvement avec perte. Pour assurer la scalabilité en mouvement, il est nécessaire d'estimer et transmettre un mouvement pour chaque résolution spatio-temporelle.

En revanche, l'indépendance du mouvement et de la décomposition ondelettes dans le cas des schémas par analyse-synthèse, permet d'estimer un seul mouvement (un seul modèle 3D) pour le GOF. Comme le mouvement n'est pas utilisé pour décorréler les informations, il est en effet possible de représenter et d'encoder le mouvement sur plusieurs niveaux de résolution indépendamment des coefficients d'ondelette. Décoder le mouvement avec perte permet d'allouer plus de débit au décodage des coefficients d'ondelettes. Si le mouvement n'est pas décodé du tout, la texture de la vidéo reconstruite a une qualité élevée mais son mouvement n'est pas reconstruit. En allouant de plus en plus de débit au mouvement, on le reconstruit peu à peu. Selon la résolution spatiale du décodeur, certaines couches de rehaussement peuvent être tronquées sans impact visuel sur la qualité du rendu. La scalabilité du mouvement est exploitée dans la thèse de Cammas [Cam04b]. Remarquons qu'une difficulté majeure est d'évaluer l'impact d'une perte en mouvement sur la qualité visuelle de la séquence synthétisée. Malgré cette difficulté, les travaux que nous avons menés dans le cadre de cette thèse s'inscrivent dans la continuité de ceux de Cammas.

Conclusion

Ce chapitre était dédié aux outils antérieurs qui permettent de modéliser, d'estimer et d'exploiter le mouvement dans une vidéo. Plusieurs modèles de mouvements

(« Block Matching », maillage déformable, modèles hybrides...) ont ainsi été introduits. Pour chacun d'entre eux, nous avons décrit des méthodes classiques pour estimer leurs paramètres. Enfin, nous avons montré que différentes approches pouvaient être suivies pour exploiter ce mouvement dans un codeur. Le codage prédictif est l'approche la plus classique choisie notamment dans les standards. En marge du codage prédictif, d'autres approches existent. Nous avons par exemple évoqué les approches par analyse-synthèse s'appuyant sur une modélisation 3D ou la création d'une mosaïque d'une scène statique. Nous avons ensuite présenté des schémas par analyse-synthèse basés sur une compensation en mouvement global d'un groupe d'images [TZ94a, WXCM99, Cam04b].

Nos travaux s'inscrivent dans la continuité de la thèse de Cammas [Cam04b, CP03b]. Comme expliqué au paragraphe 3.3.3.3, le schéma de Cammas et Pateux a pour but d'aligner les images d'un GOF sur un même instant de projection afin de les adapter à une décomposition temporelle « en ligne ». Après décomposition temporelle, les images de résidus sont envoyées au codeur JPEG2000 qui réalise une transformée ondelettes horizontale-verticale avant d'encoder les coefficients. Or, nous avons montré au chapitre 1 que l'ondelette séparable classique n'était pas adaptée au contenu géométrique d'une image fixe. Notre objectif est donc de prendre en compte la géométrie des images pour améliorer le codeur par analyse-synthèse temporelles précédent.

Dans le chapitre suivant, nous nous concentrons sur l'image fixe et présentons un schéma par analyse-synthèse qui s'inspire fortement des travaux réalisés sur le mouvement. Nous choisissons ainsi de modéliser la géométrie par un maillage déformable. A la manière d'une approche de type Bandelettes où le contenu d'un bloc de l'image est adapté à une décomposition horizontale-verticale par ondelettes, nous utilisons le maillage déformable pour adapter l'image à la décomposition séparable standard. Le critère d'adaptation utilisé pour estimer le maillage est le coût de codage de l'image déformée dans une base d'ondelettes standard. Nous proposons une technique d'optimisation très similaire à une estimation de mouvement. Après avoir encodé et transmis l'image déformée et le maillage, l'image d'origine peut être reconstruite en inversant la déformation. Deux difficultés sont à étudier. Tout d'abord la réduction du coût de codage de l'image compense-t-elle le coût du maillage à transmettre ? Ensuite, les pertes numériques introduites lors des deux ré-échantillonnages successifs ont-elles un impact sur la qualité visuelle des images ? Nous répondons à ces questions au chapitre suivant. Au chapitre 5, nous montrerons comment intégrer notre schéma par analyse-synthèse spatiales au schéma par analyse-synthèse temporelles de Cammas et Pateux.

Chapitre 4

Codage d'images fixes par adaptation du contenu spatial

Dans ce chapitre, nous proposons une nouvelle solution pour le codage d'images fixes. Cette solution tente d'apporter une réponse à trois défis principaux : la déformation de l'ondelette en fonction de la géométrie d'une image, la conservation des propriétés de l'ondelette et la « scalabilité » en géométrie. La méthode s'inspire fortement des schémas de type analyse-synthèse [TZ94a, WXCM99, Cam04b] présentés à la fin du chapitre précédent. De la même manière que ces auteurs proposent de compenser les images d'un GOF pour aligner les trajectoires de *mouvement* le long de l'axe temporel, nous proposons de réaliser une *compensation en géométrie* d'une image fixe pour adapter son contenu à un filtrage horizontal et vertical, à l'aide d'une ondelette séparable par exemple. Tout comme pour les techniques par blocs déformés, ce procédé revient à adapter l'ondelette dans le domaine image. Cependant, dans la technique proposée, la déformation est *globale* sur tout le domaine image et l'image compensée peut ensuite être encodée par JPEG2000 pour tirer partie de toutes les options du codeur. L'objectif de *géométrie « scalable »* est atteint en modélisant la compensation par un *maillage déformable*. La stratégie adoptée n'est bien sûr pas sans faille car la compensation d'une image nécessite un ré-échantillonnage irréversible.

Quatre sections composent ce chapitre. Dans un premier temps, nous identifions les briques de base de la méthode en décrivant le schéma général. Dans la seconde section, nous décrivons le critère utilisé pour adapter l'image déformée au noyau d'ondelettes et expliquons comment ce critère peut être minimisé. La troisième section est dédiée aux résultats de compression obtenus avec le schéma de base. Enfin, dans la dernière partie, nous nous penchons sur le problème des pertes liées au ré-échantillonnage de l'image et décrivons les différentes méthodes que nous avons testées pour tenter de le résoudre.

4.1 Schéma proposé

4.1.1 Principe général

Au chapitre 2, nous avons vu qu'une approche possible au problème d'adaptivité consiste à rectifier le contenu d'une image pour l'adapter à l'ondelette. Notre solution adopte cette même approche du problème mais se distingue principalement des techniques proposées par Taubman et Zakhor [TZ94b] et Le Pennec et Mallat [PM05] par les deux propriétés suivantes :

- La déformation de l'image ne se fait plus bloc par bloc mais *de façon continue* sur tout le domaine. Après déformation, il n'est plus nécessaire de modifier l'ondelette aux frontières des blocs (comme décrit au chapitre 2, paragraphe 2.3.3) pour gérer les effets de bords. Dans la technique proposée, *le filtrage se fait de manière identique sur tout le support de l'image déformée.* En particulier, l'image déformée peut être envoyée directement à un codeur ondelette existant, comme JPEG2000 [SCE01], et ainsi bénéficier de toutes ses fonctionnalités (génération d'un flux scalable, sélection d'une zone d'intérêt,...).
- Dans les deux techniques antérieures citées plus haut, un ensemble de géométries candidates est estimé pour un bloc donné en s'appuyant essentiellement sur le gradient de l'image. Or, le gradient est une donnée très bruitée. Pour extraire des trajectoires régulières, ceci contraint les auteurs à effectuer un lissage préalable de l'image. D'autre part, les critères utilisés pour calculer les paramètres géométriques ne correspondent pas à une modélisation du coût de codage des coefficients d'ondelette après décomposition. Dans la technique que nous proposons, le calcul de la déformation n'utilise pas d'a priori géométrique comme la direction du gradient dans le domaine image. *Il s'appuie sur une modélisation du coût de description de l'image déformée dans une base d'ondelettes.*

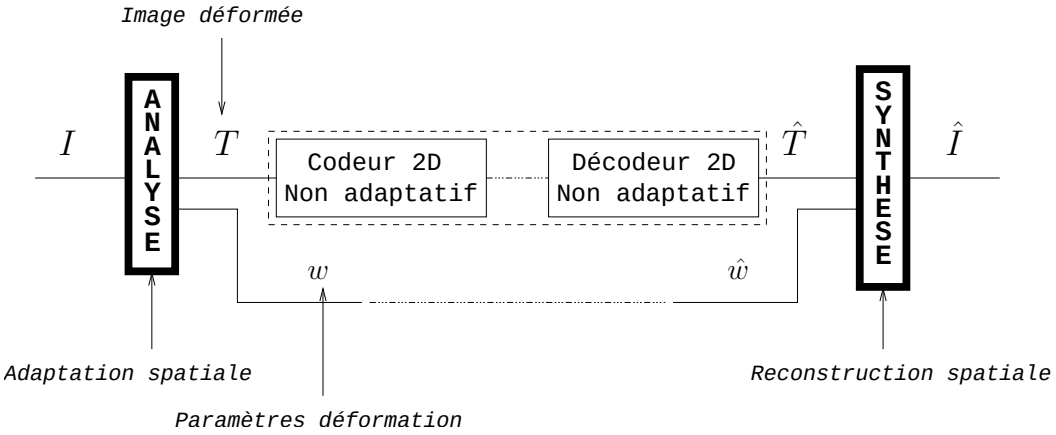


Fig. 4.1 : Méthode par Analyse-Synthèse 2D. L'image I en entrée est adaptée à un codeur image par une déformation spatiale.

Le principe général de l'approche est schématisé sur la figure 4.1. Le schéma s'appuie sur un codeur d'images fixes existant. Une brique d'*analyse* agit comme un prétraitement sur l'image I à encoder. Elle recherche une version déformée de I mieux adaptée au codeur au sens d'un certain critère $\mathbf{C}$. Cette version déformée sera notée T et sera appelée *texture*. La texture T est envoyée au codeur, transmise, puis décodée. Une brique de *synthèse* est alors placée comme post-traitement au décodeur. Elle prend en entrée la texture décodée $\hat{T}$. Nous travaillons sous l'hypothèse que la transformation spatiale w calculée à l'analyse est *inversible*. Le but de la synthèse est alors d'inverser la déformation effectuée à l'analyse pour reconstruire une version décodée $\hat{I}$ de l'image. Pour pouvoir synthétiser $\hat{I}$, il faut transmettre les paramètres de la transformation spatiale w calculés à l'analyse.

Les travaux présentés dans la suite se concentrent principalement sur les codeurs ondelettes, mais ce principe très général peut être appliqué à d'autres codeurs. Le critère $\mathbf{C}$ que nous proposons pour calculer la transformation w ne s'appuie sur aucun a priori géométrique. Pour autant, puisque w est censée « rectifier » l'image pour l'adapter à l'ondelette, on peut légitimement supposer qu'elle possède certains attributs géométriques. Dans la suite, nous pourrons donc faire référence à w sous le terme de *géométrie*. La déformation spatiale subie par l'image sera appelée *compensation géométrique*, en référence à la terminologie utilisée pour le mouvement.

4.1.2 Maillage 2D comme modèle de déformation

Au chapitre 3, nous avons vu qu'un maillage 2D peut modéliser un mouvement continu dans une vidéo. Pour ce faire, il « suffit » de définir les positions de ses sommets à un instant courant et de calculer les positions optimales, au sens d'un certain critère, à un instant de référence. Ce modèle de mouvement est équivalent à un modèle de transformation spatiale et peut être utilisé pour définir une déformation de l'image de référence qui vise à rectifier la trajectoire du mouvement.

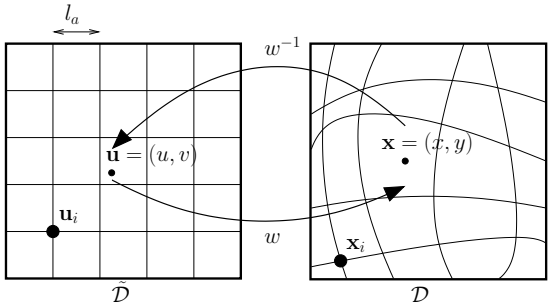


Fig. 4.2 : Maillage quadrangulaire régulier comme modèle de déformation. Le maillage est uniforme dans $\tilde{\mathcal{D}}$. Le but est de rechercher les positions optimales, au sens d'un critère $\mathbf{C}$, dans $\mathcal{D}$.

Dans notre cas, nous cherchons à rectifier une trajectoire géométrique. Supposons que la texture T existe et est connue. La problématique d'analyse peut être vue comme

une problématique d'estimation de mouvement entre T et I. Le maillage 2D apparaît donc comme un choix naturel pour modéliser une déformation inversible. En particulier, nous choisissons de recouvrir le *domaine texture* $\tilde{\mathcal{D}}$ avec un maillage $\tilde{\mathcal{M}}$ uniforme et régulier dont les sommets ont des positions fixes $\mathbf{u}_i = (u_i, v_i)$. Nous plaçons un maillage $\mathcal{M}$ de même connectivité dans le *domaine image* $\mathcal{D}$ dont les sommets sont libres de se déplacer (voir figure 4.2). Grâce à l'uniformité et à la régularité de $\tilde{\mathcal{M}}$, les seuls paramètres de la transformation spatiale sont : le nombre de sommets du maillage N_s donné par la longueur l_a d'une arête dans $\tilde{\mathcal{D}}$ et les positions des sommets $\mathbf{x}_i = (x_i, y_i)$ dans $\mathcal{D}$. Ces paramètres devront être transmis pour pouvoir reconstruire l'image en bout de chaîne.

Les positions des sommets dans $\mathcal{D}$ étant connus, tout point $\mathbf{u} = (u, v)$ du domaine texture peut être mis en correspondance avec un point $\mathbf{x} = (x, y)$ du domaine image en interpolant les positions des sommets. Ceci revient à définir la transformation spatiale w suivante :

$$\begin{aligned} w : \quad & \tilde{\mathcal{D}} \to \mathcal{D} \\ & (u, v) \mapsto (x, y) = \sum_{i=1}^{N_s} (x_i, y_i) \cdot \phi(u - x_i, v - v_i) \end{aligned} \tag{4.1}$$

où ϕ est une fonction de forme 2D définie dans $\tilde{\mathcal{D}}$. Les transformations locales possibles dépendent de la forme des facettes et de la fonction de forme choisies. Dans notre étude, nous avons fait le choix de travailler avec des mailles quadrangulaires car elles nous paraissent plus à même de capturer la régularité d'une courbe. D'autre part, la fonction ϕ choisie est la fonction bilinéaire qui limite la complexité tout en permettant une bonne variété de déformations.

Le choix du maillage parmi les modèles de géométrie possibles est particulièrement motivé par sa propriété multi-résolution intrinsèque. Les paramètres d'un maillage peuvent être encodés sous la forme d'un flux emboîté possédant une scalabilité spatiale et en qualité. Dans [Cam04b], Cammas exploite cette propriété pour proposer un schéma de codage vidéo entièrement scalable, où le mouvement peut être décodé avec pertes pour affecter plus de débit au décodage des textures. Nous verrons dans la suite si appliquer cette démarche à la géométrie apporte un gain sur une image fixe.

Comme nous l'avons vu au chapitre 2, d'autres méthodes adaptatives [LW95, Lec99b, Mar00, DDI06] utilisent le maillage 2D comme modèle géométrique. Notre solution se distingue de ces méthodes car elle n'utilise pas le maillage 2D comme grille d'échantillonnage mais comme grille de déformation. Dans notre cas, si le nombre de sommets N_s est nul, l'image est tout simplement représentée par ses coefficients d'ondelettes. Par contre, si le maillage est vu comme une grille d'échantillonnage, $N_s = 0$ signifie que l'on ne reconstruit rien. Remarquons qu'il existe tout de même un lien entre les deux approches : lorsque notre texture T est approximée avec une basse fréquence de taille $N_s \times N_s$, chaque sommet est alors rattaché à une seule intensité ; ce qui revient à approximer I avec des éléments finis comme dans les autres méthodes.

Pour conclure ce paragraphe, remarquons que le domaine texture utilisé dans nos travaux correspond au concept de domaine maître introduit par Lee et Wang [LW95] et décrit au chapitre 2. Nous avons choisi d'utiliser le terme *texture* en référence aux techniques de synthèse 3D : de la même façon qu'un objet 3D est synthétisé en plaquant une texture sur un maillage 3D, une image I est ici synthétisée en bout de chaîne en plaquant une texture sur un maillage 2D. Précisons que ce terme de texture ne doit pas être confondu avec le concept de zones texturées introduit au début de ce manuscrit. A la fin de l'analyse, la texture T contient des zones texturées mais aussi des zones homogènes et des contours (idéalement alignés le long de l'axe horizontal ou vertical).

4.1.3 Déformation image versus déformation ondelette

Considérons une transformation spatiale inversible w quelconque qui associe tout point du domaine texture à un point du domaine image. Considérons également une ondelette $\psi_{j,\mathbf{m}}$ définie sur le domaine texture $\tilde{\mathcal{D}}$, où j est le facteur d'échelle (dilatation) et $\mathbf{m}$ le facteur de translation de l'ondelette mère ψ.

Déformation de l'image. A partir de la transformation spatiale w, on peut définir un opérateur de déformation $\mathbf{W}$ qui agit sur l'image I pour donner une image déformée T dans le domaine texture :

$$T(u,v) = \mathbf{W}I(u,v) = I(w(u,v))) \quad \forall (u,v) \in \tilde{\mathcal{D}} \tag{4.2}$$

$\mathbf{W}$ est l'opérateur de compensation géométrique. Comme w est réversible, l'image I *comme fonction de* $\mathbb{R}^2$ peut être reconstruite à partir de T :

$$I(x,y) = T(w^{-1}(x,y)) = I(w \circ w^{-1}(x,y)) \quad \forall (x,y) \in \mathcal{D} \tag{4.3}$$

Déformation de l'ondelette. A partir de la transformation spatiale inverse w^{-1} , on peut définir un nouvel opérateur de déformation $\mathbf{W}^*$ qui agit cette fois sur l'ondelette $\psi_{j,\mathbf{m}}$ pour donner une ondelette déformée dans le domaine image $\mathcal{D}$:

$$\mathbf{W}^*\psi_{j,\mathbf{m}}(x,y) = \psi_{j,\mathbf{m}}(w^{-1}(x,y)) \quad \forall (x,y) \in \mathcal{D} \tag{4.4}$$

Généralement, les méthodes adaptatives [CG05, VBLVD06, WZVS06, DWW+07] se focalisent sur l'opérateur $\mathbf{W}^*$: le but est de déformer l'ondelette pour que son support capture la géométrie de l'image. C'est la solution 1 dans la figure 4.3. En se plaçant dans un domaine image continu, adapter l'ondelette à un contour revient à satisfaire les trois objectifs suivant :

1. Orienter les deux axes de l'ondelette dans les directions parallèle et orthogonale au contour,
2. Étirer le support de l'ondelette dans la direction parallèle pour tirer partie de la régularité le long du contour,

3. Contracter le support dans la direction orthogonale pour limiter les rebonds autour de la discontinuité.

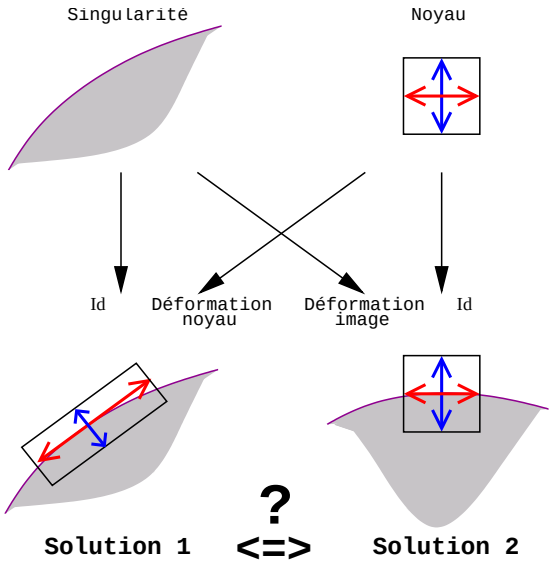


Fig. 4.3 : Deux approches au problème d'adaptativité. Solution 1 : déformer le noyau pour l'adapter à la géométrie. Solution 2 : déformer l'image pour l'adapter au noyau.

Notre solution se focalise sur l'opérateur **W** : le but est de déformer l'image pour l'adapter à l'ondelette. C'est la solution 2 dans la figure 4.3. Si on se réfère à des a priori géométriques, adapter un contour à l'ondelette séparable signifie trois choses :

1. Orienter sa direction de régularité sur un des axes (horizontal ou vertical) de l'ondelette,
2. Contracter le contour dans sa direction de régularité pour réduire le nombre de coefficients d'ondelette nécessaire à sa représentation,
3. Étirer le contour dans sa direction orthogonale pour rendre la fonction plus régulière.

Nous observons qu'il existe une certaine similarité entre les objectifs recherchés par ces deux solutions. **Mais y a-t-il *équivalence* entre les deux méthodes ?** Pour répondre à cette question, écrivons la projection de la texture T sur une ondelette $\psi_{j,\mathbf{m}}$:

$$\begin{aligned} < T, \psi_{j,\mathbf{m}} > &= \int_u \int_v T(u,v)\psi_{j,\mathbf{m}}(u,v)\mathrm{d}u\mathrm{d}v \\ &= \int_u \int_v I(w(u,v))\psi_{j,\mathbf{m}}\mathrm{d}u\mathrm{d}v \end{aligned} \tag{4.5}$$

En opérant le changement de variable $(u,v) = w^{-1}(x,y)$, on obtient :

$$< T, \psi_{j,\mathbf{m}} >= \int_x \int_y I(x,y)\mathbf{W}\psi_{j,\mathbf{m}}(x,y)J_{w^{-1}}(x,y)\mathrm{d}x\mathrm{d}y \tag{4.6}$$

où $J_{w^{-1}}(x,y)$ est le jacobien de la transformation w^{-1}. D'après la relation 4.6, la projection de T sur une ondelette est *égale* à la projection de I sur une ondelette déformée si et seulement si la condition $J_{w^{-1}}(x,y) = 1$ est respectée sur le support de l'ondelette. D'autre part, on remarque que décomposer la texture sur une base d'ondelettes non adaptatives est *équivalent* à décomposer l'image sur la base d'ondelettes déformées si et seulement si le jacobien est constant sur le support des ondelettes à chaque échelle. Dans les travaux de Taubman et Zakhor [TZ94b] et Le Pennec et Mallat [PM05], chaque bloc est déformé en réalisant des translations de lignes et/ou de colonnes. Dans ce cas, le jacobien est unitaire et la condition est donc respectée *tant que le support de l'ondelette reste à l'intérieur du bloc*. Ceci suppose en particulier que l'échelle 2^j maximale de l'ondelette soit inférieure à la taille du bloc. Forcer la valeur du jacobien à 1 limite les déformations possibles de l'image et de l'ondelette : parmi les trois objectifs donnés plus haut, seul l'objectif d'orientation peut être atteint. Pour modifier le ratio d'aspect de l'ondelette, il est alors nécessaire de recourir à un traitement spécifique après projection du bloc déformé sur la base d'ondelettes séparable, comme par exemple une « Bandelettisation » (chapitre 2, paragraphe 2.2.4.3).

Dans notre étude, la transformation spatiale w est modélisée par un maillage quadrangulaire régulier comme décrit précédemment. Avoir des mailles quadrangulaires facilite l'interprétation visuelle car la déformation de l'ondelette dans le domaine image est directement donnée par la déformation d'une maille. Dans ce cas, le choix d'une fonction de forme affine dans l'équation (4.1) permettrait d'avoir un jacobien constant et de respecter la contrainte d'équivalence à l'intérieur d'une maille tant que l'échelle 2^j de l'ondelette reste inférieure à l_a (la taille d'une arête dans $\tilde{\mathcal{D}}$). Cependant, les déformations de l'image permises par une transformation affine restent limitées. Nous lui avons donc préféré la transformation bilinéaire. Dans ce cas, le jacobien $J_{w^{-1}}$ n'est pas constant à l'intérieur d'une maille et il n'y a donc plus équivalence entre les deux techniques. La projection de T sur une ondelette revient alors à projeter I sur une ondelette déformée dont les valeurs sont pondérées par le jacobien.

4.1.4 Discrétisation de la transformée

Lorsqu'on raisonne en discret, déformer un signal est synonyme de ré-échantillonnage. Si la transformation w consiste en une translation soit des lignes soit des colonnes, comme dans [TZ94b, PM05], alors le ré-échantillonnage est *théoriquement* réversible avec une implémentation récursive d'un filtre à réponse impulsionnelle infinie. En pratique, on peut obtenir une erreur de reconstruction arbitrairement petite en augmentant le nombre de récursions du filtre et en améliorant la précision de ses coefficients. Notons d'une part que le nombre de récursions du filtre joue bien sûr sur la complexité de la déformation inverse et d'autre part que la précision des coefficients est limitée par les capacités de la machine.

Dans notre cas, l'image peut subir localement tout type de déformations selon le déplacement des nœuds dans $\mathcal{D}$: translations, rotations, étirements, contractions. Si une translation pure peut être considérée comme quasi-réversible, tous les autres types de déformations impliquent des pertes de hautes fréquences lors du passage du domaine image au domaine texture. En particulier, dans le cas d'une contraction de l'image, la zone déformée est représentée avec moins d'échantillons dans $\tilde{\mathcal{D}}$ que dans $\mathcal{D}$. On parlera de *perte de résolution*. Cette perte de résolution s'accompagne nécessairement d'un filtrage passe-bas. Dans le cas d'un étirement, la zone est représentée avec plus d'échantillons dans $\tilde{\mathcal{D}}$ que dans $\mathcal{D}$. On parlera de *gain de résolution*. Si l'étirement consiste en un sur-échantillonnage, il peut être inversé. Sinon, certains bruits seront filtrés. La fréquence de coupure dans le cas d'une perte ou d'un gain de résolution est directement liée au nombre d'échantillons dans $\tilde{\mathcal{D}}$. Comme nous l'avons vu au chapitre 3, paragraphe 3.1.4.2, la valeur du jacobien de w donne une indication sur les changements de résolution locaux. $|J_w| > 1$ se traduit par une perte de résolution. $|J_w| < 1$ se traduit par un gain de résolution. Même si des cas particuliers existent, $|J_w| = 1$ signifie dans le cas général qu'il n'y a ni gain ni perte de résolution.

En sachant que le ré-échantillonnage de l'image n'est pas réversible, nous veillerons dans la suite à analyser l'impact des pertes sur l'image reconstruite en bout de chaîne en fonction du débit : l'impact sur une métrique d'évaluation objective comme le PSNR, mais surtout l'impact visuel.

Dans la section suivante, nous nous intéressons au calcul de la déformation, c'est à dire à la brique d'analyse. Nous proposons une minimisation énergétique basée sur une modélisation du coût de description de la texture T. Nous allons ainsi voir qu'il est possible de calculer un maillage conforme aux a priori géométriques sans intégrer ces a priori au modèle. Dans la section 1.4, nous nous intéressons au codage des informations (image déformée et déformation) et montrons l'influence des paramètres d'analyse et du pas de quantification du maillage. L'étude montre qu'un maillage avec une taille d'arête $l_a = 8$ occupe une part trop importante du débit. Nous donnons alors les résultats de compression obtenus en utilisant une taille $l_a = 16$ et les comparons avec ceux fournis par JPEG2000. Lorsque le contenu géométrique de l'image reste simple, une amélioration de la qualité visuelle est observée au niveau des contours. Cependant deux difficultés sont soulevées. Tout d'abord les ré-échantillonnages successifs effectués lors de l'analyse puis de la synthèse produisent un flou d'interpolation dans les zones texturées, gênant surtout dans les hauts débits. Ensuite, une taille $l_a = 16$ ne permet pas de s'adapter aux contenus géométriques complexes que l'on trouve généralement dans les images naturelles. La section 4.4 se concentre sur ces deux difficultés. Nous proposons tout d'abord deux techniques simples pour améliorer la qualité de l'image synthétisée dans les hauts débits. Nous proposons ensuite une approche de type Quadtree permettant de raffiner la taille des mailles dans les régions de contours tout en contrôlant le coût de la géométrie. De nouvelles comparaisons avec JPEG2000 sont fournies.

4.2 L'analyse : estimation de la déformation

Rappelons que le but de la brique d'analyse est de déformer l'image pour l'adapter à une décomposition dans une base donnée. Nous nous sommes concentrés sur la base d'ondelettes séparables. Le problème peut être formulé comme la recherche des positions $\{\mathbf{x_i} = (x_i, y_i), i = 1 \dots N_s\}$ des sommets du maillage dans $\mathcal{D}$ permettant la meilleure adaptation de la texture au sens d'un critère $\mathbf{C}$. Différentes méthodes [TV91, LW95, JCLB01] ont été proposées au chapitre 2 pour construire un maillage quadrangulaire adaptatif dans $\mathcal{D}$. Cependant, leur but est de trouver une meilleure approximation de l'image par éléments finis et nous avons vu que cette approche était sensiblement différente de la nôtre.

Dans [ACSD+03], Alliez et al. proposent un remaillage anisotropique d'une surface 3D avec des mailles à dominance quadrangulaire. Leur idée est d'intégrer des lignes à partir des champs de courbure principale et secondaire de la surface. L'espacement entre deux lignes de courbure principale (respectivement secondaire) est inversement proportionnel à la courbure secondaire (resp. principale) de sorte que le ré-échantillonnage final est bien adapté à la géométrie de la surface. La première piste que nous avons suivie a consisté à adapter cette approche à l'image en remplaçant les deux champs de courbure par des champs de gradients maximal et minimal, et en utilisant la méthode d'intégration de lignes de flux décrite dans [MAD05]. Nous résumons ces travaux en annexe A. Cette technique s'avère problématique car le gradient d'une image est une donnée très bruitée. Plus un champ de vecteur est chaotique, plus il est difficile d'intégrer des lignes de grande taille. Ceci se traduit alors par un maillage très irrégulier avec beaucoup de facettes non quadrangulaires. Pour avoir des champs de vecteurs exploitables, il faut recourir à un fort lissage de l'image, ce qui limite fortement la qualité du modèle géométrique. En outre, il est impossible de prévoir le résultat fourni par cette méthode en termes de connectivité.

Dans la suite, nous contraignons la régularité du maillage et fixons arbitrairement la taille l_a d'une arête dans $\widetilde{\mathcal{D}}$. Nous proposons ensuite de résoudre le problème de l'analyse avec un modèle d'estimation itératif. A chaque itération k, nous proposons de modifier de façon conjointe la déformation $w^{(k)}$ et la texture $T^{(k)}$ pour améliorer le critère d'adaptation $\mathbf{C}^{(k)}$. A l'état initial, la déformation est l'identité, c'est-à-dire que l'on a $\mathbf{x}_i^{(0)} = \mathbf{u}_i^{(0)} \quad \forall i \in \{1 \dots N_s\}$, ou encore $T^{(0)} = I$. Il reste à définir le critère d'adaptation de la texture.

4.2.1 Coût de description texture

4.2.1.1 Définition du critère

Décomposons la texture T dans une base d'ondelettes, pour un niveau de décompostion $J \geqslant 1$ fixé. T peut être écrite comme (voir chapitre 1, paragraphe 1.3.5) :

$$T(u,v) = \sum_{\mathbf{m}} a_J[\mathbf{m}]\phi^*_{J,\mathbf{m}}(u,v) + \sum_{j=1}^{J}\sum_{\mathbf{m}} d_j[\mathbf{m}]\psi^*_{j,\mathbf{m}}(u,v) \tag{4.7}$$

où $\mathbf{m}$ parcourt tous les échantillons d'une sous-bande j donnée. L'énergie de la texture est donc répartie sur la basse fréquence J et sur un ensemble de détails $\{d_j[\mathbf{m}]\}_{j,\mathbf{m}}$. Puisque nous travaillons dans un cadre de compression, le but de l'analyse est d'obtenir une texture qui soit plus facile à coder que l'image de départ. Nous proposons donc de formuler le critère $\mathbf{C}$ comme un coût de description de T dans la base d'ondelettes. Pour modéliser ce coût, nous faisons ici l'hypothèse qu'un détail dans une sous-bande j donnée est une variable aléatoire D_j qui suit une loi de probabilité gaussienne centrée de variance σ_j^2, à savoir :

$$\mathbf{P}\{D_j = d\} = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp(-\frac{d^2}{2\sigma_j^2}) \tag{4.8}$$

Le coût de description des détails $\{d_j[\mathbf{m}]\}$ pour $j = 1 \dots J$ peut alors s'exprimer de la façon suivante :

$$\begin{aligned} \mathbf{C} &= -\sum_{j=1}^{J} \sum_{\mathbf{m}} \log_2 \mathbf{P}\{D_j = d_j[\mathbf{m}]\} \\ \mathbf{C} &\propto -\sum_{j=1}^{J} \sum_{\mathbf{m}} \frac{d_j[\mathbf{m}]^2}{2\sigma_j^2} + K \end{aligned} \tag{4.9}$$

où K est une constante. **$\mathbf{C}$ est le critère choisi pour estimer la déformation** w. Comme nous pouvons le voir, ce critère ne prend en compte aucun a priori sur la géométrie. Cependant, réduire $\mathbf{C}$ revient à déplacer l'énergie du signal vers les basses fréquences et donc à en construire une représentation creuse. Un grand nombre de détails nuls signifie que l'ondelette capture bien le contenu de la texture. Alternativement, cela signifie que la déformation *adapte T à l'ondelette*. On s'attend donc à ce que la déformation et la texture obtenues satisfassent certains a priori géométriques.

Sous la forme (4.9), le coût de description n'est pas défini comme une fonction de w. On peut néanmoins rechercher un minimum de $\mathbf{C}$ de façon exhaustive en déplaçant localement les sommets $\mathbf{x}_i$ un par un (à la manière de l'estimation de mouvement vue au chapitre 3, paragraphe 3.2.1). Ceci s'avérerait cependant très lourd car pour chaque nouveau déplacement d'un nœud, il faudrait recalculer la décomposition de T dans la base d'ondelettes. Notre but est d'exprimer $\mathbf{C}$ comme une fonction de w pour pouvoir faire une recherche globale des meilleures positions, par exemple par une technique de descente du gradient de $\mathbf{C}(w)$.

L'hypothèse d'une loi gaussienne dans les sous-bandes d'ondelettes peut être discutée. Une loi laplacienne ou une mixture de gaussiennes généralisées peuvent en effet être considérées comme des modèles plus justes pour des images naturelles. La loi gaussienne a été choisie principalement car elle conduit à une expression quadratique facilement dérivable.

4.2.1.2 Expression en fonction de w

Dans la suite, nous utiliserons la notation $\tilde{T}_j$ pour désigner l'approximation de T à l'échelle 2^j :

$$\tilde{T}_j(u,v) = \sum_{\mathbf{m}} a_j[\mathbf{m}]\phi^*_{j,\mathbf{m}}(u,v) \tag{4.10}$$

La figure 4.4 montre quelques approximations de T à l'état initial où l'on a $T = I$.

$\tilde{T}_2$ $\tilde{T}_1$ $\tilde{T}_0 = T$

Fig. 4.4 : $\tilde{T}_j$ est l'approximation de T obtenue en mettant à 0 tous les détails $d_k[\mathbf{m}]$ pour $k \in \{1..j\}$.

A partir de l'expression (4.7), nous pouvons écrire :

$$\sum_{j=1}^{J}\sum_{\mathbf{m}} d_j[\mathbf{m}]\psi^*_{j,\mathbf{m}}(u,v) = T(u,v) - \tilde{T}_J(u,v) \tag{4.11}$$

L'application du théorème de Parseval fournit l'égalité suivante :

$$\sum_{j=1}^{J}\sum_{\mathbf{m}} d_j[\mathbf{m}]^2 = \sum_{(u,v)\in\tilde{\mathcal{D}}} (T(u,v) - \tilde{T}_J(u,v))^2 \tag{4.12}$$

Comme $T = I(w(u,v))$, l'égalité précédente montre que l'énergie des hautes fréquences peut être exprimée en fonction de la déformation et de l'approximation $\tilde{T}_J$. Nous souhaitons établir une expression similaire pour le coût de description $\mathbf{C}$. La définition (4.9) de $\mathbf{C}$ faisant intervenir des poids associés à chaque échelle 2^j, son expression en fonction de w nécessite un développement supplémentaire. Pour simplifier les notations, nous noterons $\sum_{(u,v)\in\tilde{\mathcal{D}}}(T(u,v) - \tilde{T}_j(u,v))^2 = \sum_{\tilde{\mathcal{D}}}(T - \tilde{T}_j)^2$ pour une échelle j donnée.

Considérons un ensemble de poids $\{\eta_j \quad \backslash \quad j = 1 \ldots J\}$. D'après l'équation (4.12), on peut écrire :

$$\begin{aligned}
\eta_1 \sum_{\tilde{\mathcal{D}}} (T - \tilde{T}_1)^2 &= \eta_1 \sum_{\mathbf{m}} d_1[\mathbf{m}]^2 \\
\eta_2 \sum_{\tilde{\mathcal{D}}} (T - \tilde{T}_2)^2 &= \eta_2 (\sum_{\mathbf{m}} d_1[\mathbf{m}]^2 + \sum_{\mathbf{m}} d_1[\mathbf{m}]^2) \\
&\vdots \\
\eta_J \sum_{\tilde{\mathcal{D}}} (T - \tilde{T}_J)^2 &= \eta_J (\sum_{\mathbf{m}} d_1[\mathbf{m}]^2 + \cdots \sum_{\mathbf{m}} d_J[\mathbf{m}]^2)
\end{aligned} \tag{4.13}$$

En sommant toutes les lignes, on arrive à l'égalité suivante :

$$\sum_{j=1}^{J} \eta_j \sum_{\tilde{\mathcal{D}}} (T - \tilde{T}_j)^2 = \sum_{j=1}^{J} \varsigma_j \sum_{\mathbf{m}} d_j[\mathbf{m}]^2 \tag{4.14}$$

avec

$$\varsigma_j = \sum_{k=1}^{j} \eta_k \quad \forall j \in \{1 \dots J\} \tag{4.15}$$

En remplaçant ς_j par $\frac{1}{2\sigma_j^2}$, le terme de droite dans l'équation (4.14) correspond au coût de description $\mathbf{C}$ à une constante près. En inversant le système (4.15), on détermine la valeur des poids η_j :

$$\begin{cases} \eta_J = \varsigma_J = \frac{1}{2\sigma_j^2} \\ \eta_j = \varsigma_j - \varsigma_{j+1} = \frac{1}{2\sigma_j^2} - \frac{1}{2\sigma_{j+1}^2} \quad \forall j \in \{1 \dots J-1\} \end{cases}$$

et le coût de description de la texture $\mathbf{C}$ peut finalement être exprimé en fonction de la transformation w et des approximations $\tilde{T}_j$ pour $j \in \{1 \dots J\}$:

$$\mathbf{C} = \sum_{j=1}^{J} \eta_j \sum_{\tilde{\mathcal{D}}} (I(w(u,v)) - \tilde{T}_j(u,v))^2 \tag{4.16}$$

Pour la plupart des images naturelles, on observe que la variance dans une sous-bande d'ondelettes augmente avec l'échelle 2^j. Ceci signifie en particulier que $1/\sigma_{j+1}^2 < 1/\sigma_j^2 \quad \forall j \in \{1 \dots J-1\}$ et donc les coefficients η_j sont généralement positifs.

4.2.1.3 Optimisation conjointe du couple (w, T)

Pour minimiser le coût de description sous sa forme (4.16), nous proposons de faire une recherche itérative sur w de type descente en gradient. Une telle recherche permet une optimisation globale de l'ensemble des positions $\{\mathbf{x}_i\}$. Cependant, dans l'expression de $\mathbf{C}$, nous remarquons que les approximations $\tilde{T}_j$ dépendent elles aussi de w car elles dépendent de T. Dans le cas d'une recherche exhaustive par mise en correspondance (voir chapitre 3, paragraphe 3.2.3.1), cela ne pose pas de problème car on ne cherche pas

à dériver une énergie. Dans une technique de type descente en gradient, dériver les approximations $\tilde{T}_j$ produirait une expression trop complexe à exploiter. Pour cette raison, nous avons choisi de considérer T comme une variable à part entière dans l'optimisation. Nous formulons alors la minimisation de $\mathbf{C}(w,T)$ comme un problème d'optimisation conjointe : on recherche le couple (w^*,T^*) pour lequel le coût de description est minimal.

A l'état initial, la déformation est l'identité : $w^{(0)} = Id$. Ceci signifie en particulier que $\mathbf{x}_i^{(0)} = \mathbf{u}_i^{(0)} \quad \forall i \in \{1 \dots N_s\}$, et donc que la texture est l'image : $T^{(0)} = I$. Connaissant la texture $T^{(k)}$ et la déformation $w^{(k)}$ à la fin de l'itération k, l'itération $k+1$ se déroule en deux temps.

Au premier temps, la déformation $w^{(k+1)}$ est calculée en minimisant le coût $\mathbf{C}^{(k+1)} = \mathbf{C}(w^{(k+1)})$:

$$\mathbf{C}^{(k+1)} = \sum_{j=1}^{J} \eta_j \sum_{\tilde{\mathcal{D}}} (I(w^{(k+1)}(u,v)) - \tilde{T}_j^{(k)}(u,v))^2 \tag{4.17}$$

Dans cette expression, les approximations $\tilde{T}_j^{(k)}$ sont connues et ne dépendent donc pas de $w^{(k+1)}$. Pour minimiser $\mathbf{C}^{(k+1)}$, nous proposons d'appliquer la technique d'estimation de mouvement décrite à l'annexe B. Il s'agit de calculer la dérivée de $\mathbf{C}^{(k+1)}$ par rapport à chaque position $\mathbf{x}_i^{(k+1)}$ puis de linéariser cette dérivée à la position $\mathbf{x}_i^{(k)}$. Annuler chaque dérivée produit un système linéaire de type $A \cdot \Delta\mathbf{X}^{(k+1)} = B$ dont les inconnues sont les déplacements $\Delta\mathbf{x}_i^{(k+1)}$ par rapport aux positions $\mathbf{x}_i^{(k)}$ trouvées à l'étape précédente.

Au deuxième temps, la nouvelle texture $T^{(k+1)}$ est calculée en déformant l'image d'origine I avec la déformation $w^{(k+1)}$:

$$T^{(k+1)}(u,v) = I(w^{(k+1)}(u,v)) \quad \forall (u,v) \in \tilde{\mathcal{D}} \tag{4.18}$$

et les approximations $\tilde{T}_j^{(k+1)}$ sont calculées en utilisant l'équation (4.10).

Cet algorithme d'analyse est schématisé figure 4.5. Il peut être vu comme un algorithme d'espérance-maximisation (EM). A chaque étape, une observation de la variable T est faite et la déformation est mise à jour en calculant de petits déplacements des sommets pour améliorer le coût $\mathbf{C}(T,w)$, c'est-à-dire pour améliorer l'adaptation de T à la base d'ondelettes. T est ensuite mise à jour avec la nouvelle observation de w. Dans notre cadre de travail, remarquons qu'un tel algorithme n'assure pas d'atteindre le minimum global de la fonction $\mathbf{C}(w,T)$ car cette fonction n'est en général pas convexe. On ne peut pas non plus assurer que l'algorithme converge vers un minimum (global ou local) en un petit nombre d'itérations. De ce fait, la démarche habituelle est de se fixer un nombre maximal d'itérations pour réduire $\mathbf{C}$. Ce nombre sert donc de critère d'arrêt à l'algorithme. Comme nous l'avons vu au chapitre 3, ce type d'approche a été largement utilisé et validé dans un contexte d'estimation de mouvement entre deux images d'une séquence vidéo. Nous verrons plus loin les résultats obtenus dans le cadre de notre estimation géométrique.

4.2.1.4 Simplifications

Sous la forme (4.17), nous remarquons que le coût de description peut être vu comme une somme pondérée de J différences d'images déplacées (DID). Ici, nous montrons qu'il

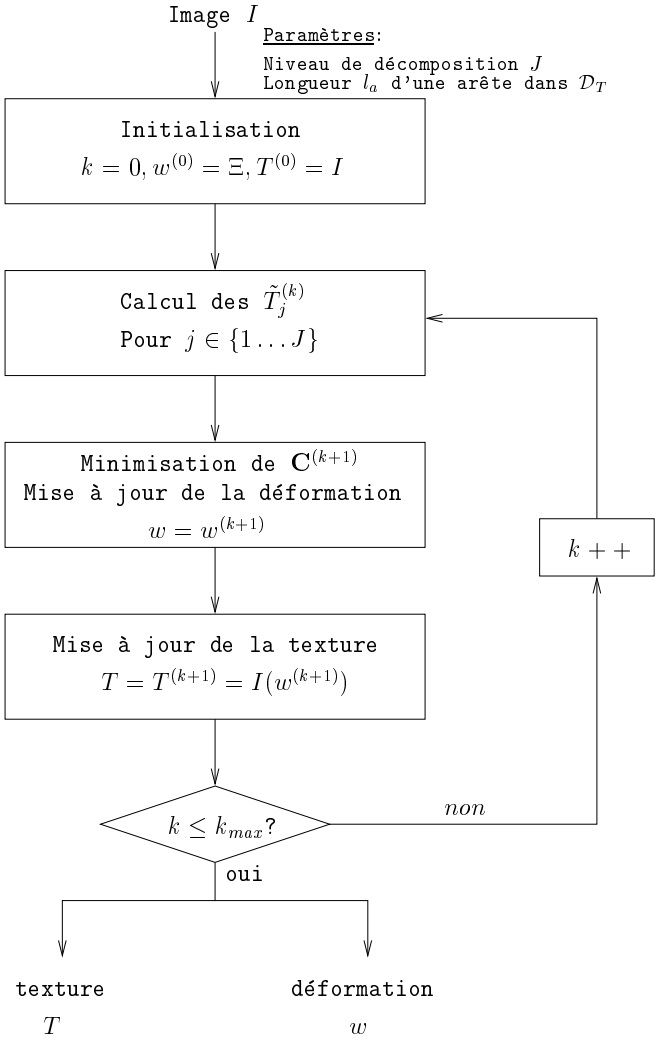


Fig. 4.5 : Schéma de l'analyse. La texture et la déformation sont générées de façon itérative en appliquant un algorithme d'espérance-maximisation.

est possible de simplifier l'expression de $\mathbf{C}$ pour aboutir à une seule DID comme dans le cas d'une estimation de mouvement. A chaque itération $k+1$, la complexité de l'analyse se répartit alors sur trois étapes : la construction du système linéaire, la résolution du système linéaire et le calcul d'une texture $T_{cible}^{(k+1)}$ appelée *texture cible* que nous dérivons plus bas.

Construction du système linéaire. Par souci de clarté, nous faisons abstraction de l'indice de l'itération et des coordonnées (u, v). Développons l'équation (4.17) :

$$\begin{aligned} \mathbf{C} &= \sum_j \eta_j \sum_{\tilde{\mathcal{D}}} (T - \tilde{T}_j)^2 \\ &= \sum_{\tilde{\mathcal{D}}} (\sum_j \eta_j T^2 - 2T \sum_j \eta_j \tilde{T}_j + \sum_j \eta_j \tilde{T}_j^2) \end{aligned} \tag{4.19}$$

En mettant en facteur le terme $\sum_j \eta_j$, on montre que :

$$\mathbf{C} = (\sum_j \eta_j) \cdot [T - T_{cible}]^2 - \frac{(\sum_j \eta_j \tilde{T}_j)^2}{\sum_j \eta_j} + \sum_j \eta_j \tilde{T}_j^2 \tag{4.20}$$

où T_{cible} est appelée *texture cible*. T_{cible} est une somme pondérée d'approximations $\tilde{T}_j$:

$$T_{cible} = \frac{\sum_j \eta_j \tilde{T}_j}{\sum_j \eta_j} \tag{4.21}$$

En rappelant que les approximations $\tilde{T}_j^{(k)}$ sont considérées comme indépendantes de la déformation $w^{(k+1)}$ lors de sa mise à jour, nous pouvons réécrire le coût (4.17) :

$$\mathbf{C}(w^{(k+1)}) = (\sum_j \eta_j) \cdot \mathbf{C}'(w^{(k+1)}) + K \tag{4.22}$$

Avec

$$\mathbf{C}'(w^{(k+1)}) = \sum_{\tilde{\mathcal{D}}} (I(w^{(k+1)}) - T_{cible}^{(k)})^2 \tag{4.23}$$

et K est un terme qui ne dépend pas de $w^{(k+1)}$.

Au final, la relation (4.22) montre que minimiser $\mathbf{C}(w^{(k+1)})$ revient à minimiser le coût $\mathbf{C}'(w^{(k+1)})$. Comme $\mathbf{C}'$ est une DID, la construction du système linéaire a désormais la même complexité que lors d'une estimation de mouvement.

Résolution du système linéaire. La matrice A du système linéaire est une matrice de taille $2N_s \times 2N_s$. Comme nous le montrons à l'annexe B, cette matrice est une matrice creuse avec seulement quelques coefficients non nuls par ligne. Le nombre de coefficients non nuls dépend de la fonction de forme utilisée dans la définition (4.1)

de la déformation. Si la fonction bilinéaire est utilisée, chaque ligne a au plus 9 coefficients non nuls, car le déplacement d'un nœud est uniquement lié aux déplacements de ses 8 voisins les plus proches. Avec une telle matrice, des méthodes très efficaces de résolution du système linéaire existent, telles que les méthodes par gradient conjugué (voir [PFTV92], chapitre 2), conduisant à une complexité quasi-linéaire par rapport au nombre de nœuds N_s.

Calcul de $T^{(k)}_{cible}$. La texture cible est calculée à la fin de chaque itération. Ce calcul nécessite la connaissance de T. La valeur de T en chaque pixel $(u, v) \in \tilde{\mathcal{D}}$ se calcule en deux temps. Dans un premier temps, le correspondant $(x, y) = w(u, v)$ de chaque pixel dans $\mathcal{D}$ est calculé. Ce correspondant est une position dans $\mathbb{R}^2$. Dans un deuxième temps la valeur $I(x, y)$ est interpolée à partir des valeurs aux pixels. La complexité du calcul de T dépend des fonctions de forme utilisées pour le calcul du correspondant et pour l'interpolation des intensités. Dans tous les résultats qui seront présentés plus loin, *nous avons utilisé une interpolation bicubique.*

Il est possible de s'épargner l'étape d'interpolation des intensités en calculant une version super-résolue de I avant toute itération. Ensuite, à chaque itération, l'échantillon le plus proche de (x, y) dans $\mathcal{D}$ est choisi (interpolation au plus proche). Plus la dimension de l'image super-résolue sera grande, plus la valeur calculée sera proche de la valeur réelle de $I(x, y)$.

Une fois $T^{(k)}$ connue, toutes ses approximations $\tilde{T}^{(k)}_j$ pour $j \in \{1 \dots J\}$ peuvent être calculées. Ceci nécessite une transformée en ondelettes de $T^{(k)}$ suivie de J transformées inverses. $T^{(k)}_{cible}$ peut ensuite être déterminée à l'aide de l'équation (4.21).

En termes de complexité globale, la seule étape qui différencie notre estimation géométrique d'une estimation de mouvement par maillage est le calcul de la texture cible à la fin de chaque itération. **Tout se passe comme si on estimait un mouvement entre l'image d'origine et une texture cible dont la connaissance se raffine à chaque itération.**

4.2.2 Conformité du maillage

La conformité du maillage est nécessaire pour définir une transformation w bijective et inversible. Il faut donc s'assurer que cette propriété reste vraie à l'issue de l'analyse. En pratique, trois « traitements » spécifiques sont mis en œuvre pour contrôler la déformation du maillage.

Tout d'abord, une augmentation de Levenberg-Marquardt est réalisée sur la diagonale de la matrice A du système linéaire. Comme expliqué à l'annexe B, cette augmentation a pour but d'éviter les grands déplacements de nœuds dans les zones où le gradient de l'image est très faible. Ensuite, une énergie de déformation $\mathbf{E}_d$, ou énergie ressort, comme celle employée dans [WL94] (voir page 94) est additionnée au coût de description $\mathbf{C}'$. Cette énergie permet de rendre la transformation plus régulière en forçant les nœuds à bouger ensemble. Dans notre implémentation, nous avons considéré toutes les constantes de ressort comme unitaires. L'énergie ressort est combinée au coût

avec un certain poids ω_d. La valeur de ce poids est un paramètre de l'analyse et son choix est le résultat d'un compromis entre adaptivité et parcimonie (plus le maillage est peu déformé, moins il coûtera cher à coder). L'énergie finale $\mathbf{E}$ que nous cherchons à minimiser est $\mathbf{E} = \mathbf{C}' + \omega_d \cdot \mathbf{E}_d$.

Les deux outils précédents permettent de contrôler la déformation mais n'empêchent nullement la dégénérescence des mailles dans les zones à fort gradient. Pour forcer chaque maille à rester conforme, nous effectuons à l'issue de chaque itération la « projection non-obtuse » introduite par Wang et Lee dans [WL94] et décrite page 96.

4.2.3 Gestion des bords

Pour éviter de mettre en place un traitement spécial pour les sommets situés à la frontière du maillage, ces sommets sont fixés au début de l'analyse et ne sont pas considérés dans la minimisation énergétique. Après chaque itération, ils peuvent être déplacés sur leur frontière respective pour satisfaire des contraintes de conformité. Pour améliorer l'adaptativité aux bords de l'image, il est possible d'utiliser un maillage de départ plus grand que le domaine image. Cependant, cette démarche n'a pas apporté de gain significatif et nous ne l'avons donc pas appliquée pour générer les résultats de ce chapitre. Par contre, nous y reviendrons dans le chapitre 5.

4.2.4 Exemples d'analyse-synthèse

Dans ce paragraphe, nous illustrons les étapes d'analyse et de synthèse en considérant l'image *Lena* 256×256. Le but est de valider la technique d'estimation géométrique proposée (problématique d'*analyse*) mais aussi d'observer la qualité de l'image obtenue après déformation inverse (problématique de *synthèse*). Dans ces exemples, aucune information n'est quantifiée ni encodée.

4.2.4.1 Efficacité de l'analyse

Pour évaluer l'efficacité de l'analyse, nous réalisons une expérience en choisissant des mailles de taille 8×8 dans $\tilde{\mathcal{D}}$ ($l_a = 8$). L'ondelette utilisée pour calculer les approximations $\tilde{T}_j$ est l'ondelette de Daubechies 9/7 [CDF89a, ABMD92]. Le niveau de décomposition J (correspondant au nombre de sous-bandes de haute fréquence prises en compte dans le coût de description) est choisi égal à 4 et l'énergie de déformation est mise à 0 ($\omega_d = 0$). Nous laissons tourner l'algorithme sur un grand nombre d'itérations ($k_{max} = 100$). La figure 4.6 montre les sorties de la brique d'analyse.

Observations sur le maillage. A droite de la figure 4.6 sont représentés le maillage $\mathcal{M}$ déformé et l'image d'origine I dans le domaine $\mathcal{D}$. Rappelons que la forme des mailles dans le domaine image indique la déformation du support de l'ondelette. Au début de l'analyse, $\mathcal{M}$ est uniforme : le support de l'ondelette est le support carré classique. A la fin de l'analyse, nous observons que les mailles se sont déformées. Au paragraphe 4.1.3, nous avions donné 3 objectifs en nous appuyant sur des a priori géométriques : l'orientation (1) et l'élongation (2) de l'ondelette dans la direction du contour, ainsi que

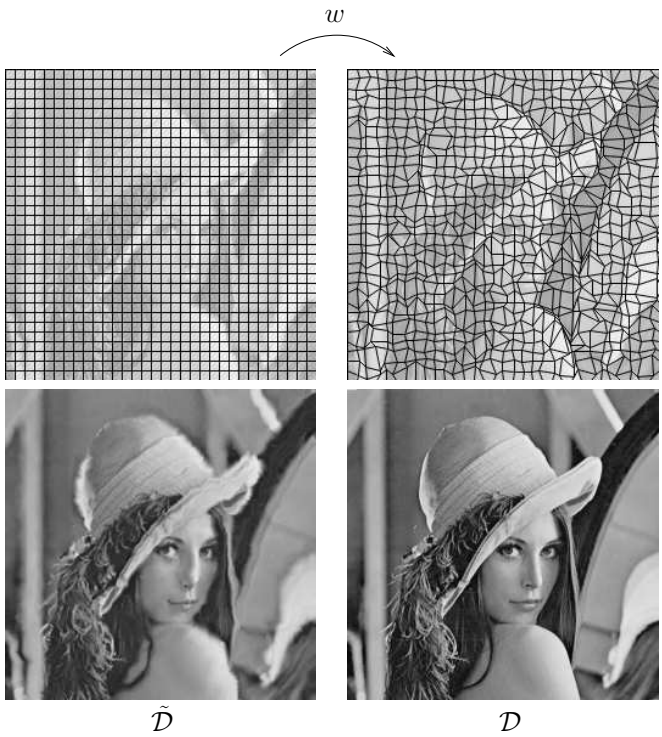


Fig. 4.6 : Un résultat d'analyse sur *Lena* 256 avec $k_{max} = 100$, $J = 4$, $l_a = 8$, $\omega_d = 0$. [A droite] Maillage dans $\mathcal{D}$ et image originale, [A gauche] Maillage dans $\tilde{\mathcal{D}}$ et texture obtenue.

sa contraction (3) dans la direction orthogonale au contour. En observant les résultats dans le domaine image, nous pouvons faire les remarques suivantes :

- Les mailles se sont resserrées autour des contours. Ceci est particulièrement significatif sur les contours du chapeau, du miroir, de l'épaule... Cette contraction des mailles revient à contracter le support de l'ondelette dans les directions orthogonales aux contours et correspond à l'objectif 3.
- Lorsque le contour d'origine est *courbe*, les mailles ont des difficultés à capturer leur orientation. Cette difficulté n'est pas liée à l'optimisation mais au modèle de géométrie choisi. En effet, toutes les mailles sont forcées à rester connectées et ne peuvent donc pas tourner librement. Ceci se traduit par des mailles en forme de losanges. On l'observe par exemple sur les contours du miroir. La contrainte de connectivité régulière empêche donc de satisfaire partout l'objectif 1. Elle empêche également de satisfaire partout l'objectif 2 car une maille qui s'étire le long d'un contour produit nécessairement la contraction d'une maille voisine.
- Remarquons enfin que les nœuds se sont également déplacés dans les zones homogènes et texturées qui ne sont pas proches d'un contour. Ces déplacements sont moins significatifs mais montrent qu'il existe une activité non isotrope du gradient dans ces zones.

Observations sur la texture. A gauche de la figure sont représentés le maillage $\tilde{\mathcal{M}}$ et la texture dans le domaine $\tilde{\mathcal{D}}$. Au paragraphe 4.1.3, nous avions donné 3 objectifs en nous appuyant sur des a priori géométriques : l'alignement des contours dans la direction horizontale ou verticale (1), la contraction du contour dans sa direction régulière (2) et l'étirement du contour dans la direction orthogonale (3). Les résultats dans le domaine texture sont cohérents avec les observations précédentes :

- La majorité des contours ont été « étirés » dans la direction orthogonale à la discontinuité. Comme nous l'avons dit, ceci revient à rendre la fonction plus régulière dans cette direction et donc plus adaptée à une décomposition par ondelettes. Ceci correspond à l'objectif 3.
- Certains contours qui n'étaient ni horizontaux ni verticaux à l'origine ont été alignés le long de l'un de ces deux axes. On remarque par exemple que le nez de *Lena* ainsi que ses cheveux (à droite) ont été alignés sur l'axe vertical dans le domaine texture. On remarque également l'apparition d'un « phénomène d'escalier » au niveau de l'épaule. Ces observations sont en accord avec l'objectif 1. L'objectif 2 qui est l'étirement des contours dans la direction régulière est plus difficilement observable.

Ainsi, même sans avoir intégré d'a priori géométrique dans notre optimisation, nous constatons que **le maillage et la texture obtenus satisfont certains de nos a priori géométriques**. Notons que parfois l'interprétation visuelle est plus difficile. La figure 4.7 montre par exemple les sorties de l'analyse en utilisant une taille d'arête $l_a = 16$. Nous voyons que les mailles capturent la géométrie de façon moins fine et que certaines caractéristiques comme le nez ne sont plus déformées. Cependant, rappelons que le but de l'optimisation, indépendamment de tout a priori géométrique, est de

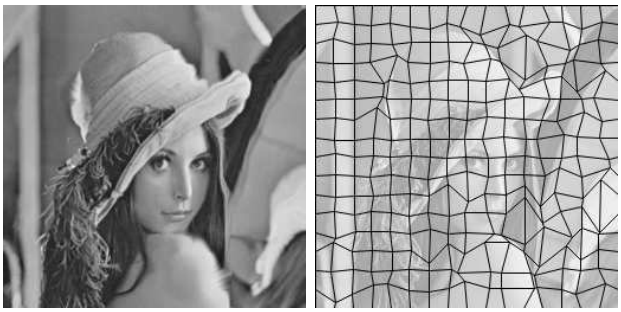


Fig. 4.7 : Un résultat d'analyse sur *Lena* 256 avec $k_{max} = 100$, $J = 4$, $l_a = 16$, $\omega_d = 0$. [A droite] Maillage dans $\mathcal{D}$ et image originale, [A gauche] Maillage dans $\tilde{\mathcal{D}}$ et texture obtenue.

réduire le coût de codage de la texture. Sur la figure 4.8, nous avons représenté l'évolution de l'énergie dans les quatre premières sous-bandes de haute fréquence et de l'énergie totale au cours des itérations, pour les cas $l_a = 8$ et $l_a = 16$. Pour ces deux cas, nous pouvons faire les mêmes observations :

- L'énergie de la première sous-bande diminue de façon très significative après une dizaine d'itérations. A la fin de l'analyse, elle est divisée par 3 dans le cas $l_a = 8$ et par plus de 2 dans le cas $l_a = 16$. L'énergie des deuxième et troisième sous-bande diminue également de façon significative mais la convergence est de plus en plus lente.
- L'énergie des quatrième et cinquième (non représentée ici) sous-bandes *augmente*. On remarque aussi que **l'énergie totale reste constante**. Ces observations sont importantes car elles indiquent que l'analyse a pour effet de *déplacer l'énergie des hautes fréquences vers les basses fréquences*. Ceci est en accord avec la recherche d'une représentation parcimonieuse de l'image.

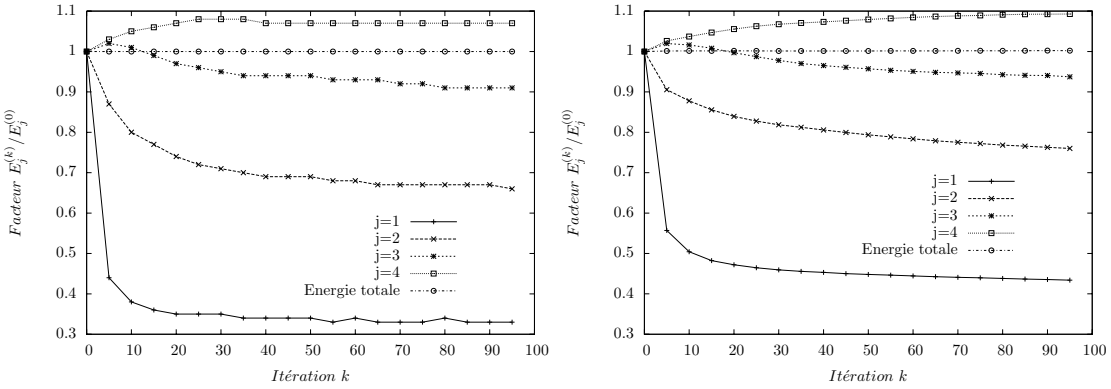


Fig. 4.8 : Evolution de l'énergie dans les sous-bandes de haute fréquence. $\mathbf{E}_j^{(k)}$ correspond à l'énergie de la sous-bande d'échelle 2^j à l'itération k. [A gauche] Avec $l_a = 8$, [A droite] Avec $l_a = 16$.

Ainsi, les remarques précédentes montrent que malgré la contrainte de connectivité régulière et des mailles de grande taille, l'optimisation proposée est efficace et satis-

fait l'objectif de départ : la texture est mieux adaptée que l'image de départ à une décomposition par ondelettes.

Notons que les différents paramètres (J, $\mathbf{E}_d$, l_a, k_{max}) peuvent modifier le résultat de l'analyse. Dans la suite, nous utiliserons un nombre d'itérations k_{max} égal à 30. Nous discuterons du choix des autres paramètres dans la section consacrée au codage car le jeu de paramètres optimal pour l'analyse n'est pas nécessairement celui qui aboutira au meilleur compromis débit-distorsion en bout de chaîne. En particulier, la taille d'une maille a beaucoup d'impact sur le coût du maillage.

4.2.4.2 Synthèse

Dans tous les raisonnements précédents, nous avons fait l'hypothèse que la transformation w était inversible. En utilisant des mailles connectées pour modéliser la transformation, cette hypothèse est vérifiée : w décrit une bijection entre le domaine texture $\tilde{\mathcal{D}}$ et le domaine image $\mathcal{D}$ et est donc inversible. L'étape de synthèse a pour but de reconstruire l'image de départ en inversant la déformation effectuée à l'analyse. Du fait des pertes dues aux ré-échantillonnages successifs lors de l'analyse et de la synthèse, l'image reconstruite n'est pas exactement égale à l'image d'origine. Dans la suite, nous noterons $I^\star$ *l'image de qualité maximale que l'on peut reconstruire sans perte sur la texture et la déformation* w. On a :

$$I^\star(x,y) = T(w^{-1}(x,y)) \quad \forall (x,y) \in \mathcal{D} \tag{4.24}$$

On remarque que l'étape de synthèse a une complexité limitée. Elle requiert la mise en correspondance de chaque pixel du domaine image avec une position du domaine texture. Connaissant la maille à laquelle un pixel donné appartient, Wang et Lee [WL96a] donnent les formules permettant de connaître son correspondant dans $\tilde{\mathcal{D}}$ dans le cas de mailles quadrangulaires. Comme ce correspondant est une position flottante, une interpolation est nécessaire pour obtenir l'intensité recherchée.

La figure 4.9 illustre les étapes d'analyse-synthèse. La qualité de $I^\star$ en bout de chaîne est importante car elle conditionne les résultats du codeur dans les hauts débits. L'image synthétisée sur cette figure a un PSNR égal à 38.02 dB. Sur la figure 4.10, nous reproduisons l'image originale et l'image synthétisée et affichons également l'image d'erreur. Visuellement, nous pouvons observer qu'un léger flou a été introduit dans l'image reconstruite par rapport à l'image originale. Le flou est surtout visible dans les zones texturées comme les plumes du chapeau ou les cheveux. Il est également visible autour des yeux de *Lena*. L'image d'erreur supporte ces observations. On remarque en effet que l'écart le plus important se situe dans les zones texturées. On remarque aussi qu'une erreur *numérique* de reconstruction existe au niveau des contours. Néanmoins, cette erreur, bien qu'elle soit prise en compte dans le calcul du PSNR, n'est guère visible à l'œil nu. Il est important de garder cet élément en tête pour pondérer les mesures en PSNR qui seront données dans la suite de ce chapitre. En particulier, dans la section suivante nous nous intéressons au codage des informations et présentons les résultats du schéma basique en termes de compression. Dans la dernière section, nous chercherons en particulier à améliorer le rendu des zones texturées.

Fig. 4.9 : Illustration des étapes d'analyse synthèse. $I^\star$ est l'image de qualité maximale qu'il est possible de reconstruire. Son PSNR est égal à 38.02 dB.

Fig. 4.10 : (a) Image originale, (b) Image reconstruite après synthèse (PSNR=38.02 dB), (c) Image de l'erreur multipliée par 10.

4.3 Compression

4.3.1 Codage de la texture et du maillage

A l'issue de l'analyse, l'image I est représentée par deux informations : la texture T et la déformation w. Ces deux informations doivent être quantifiées, codées et transmises pour pouvoir synthétiser une image $\hat{I}$ en bout de chaîne.

4.3.1.1 Codage de la texture

Rappelons que la texture T a été calculée pour s'adapter à une décomposition en ondelettes. Dans notre implémentation, le codage de T est réalisé par JPEG2000 (VM 8.0). Pour être cohérent avec l'analyse, le noyau d'ondelettes choisi dans JPEG2000 pour effectuer la transformée en ondelettes doit être le même que celui choisi lors de l'analyse. Dans les tests présentés plus bas, la base d'ondelettes choisie pour l'analyse et le codage est la base d'ondelettes de Daubechies 9/7 [CDF89a]. D'autre part, nous activons l'option `-Clayers` de JPEG2000. Cette option permet de générer un flux scalable composé de 50 couches de qualité correspondant à des débits répartis de façon logarithmique entre 0.05 bpp et 2.00 bpp. Au décodage, la texture pourra donc être décodée à différents débits à partir du même flux. Tous les autres paramètres de JPEG2000 conservent leur valeur par défaut.

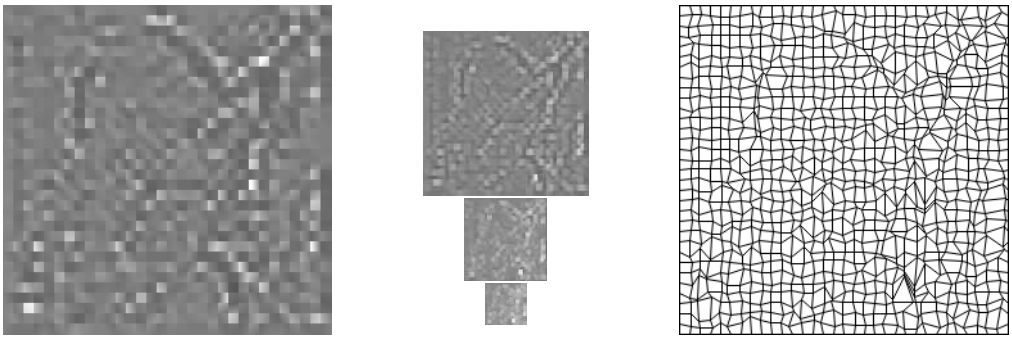
4.3.1.2 Quantification adaptative de la texture

En appliquant le schéma de principe donné figure 4.1, la quantification et l'encodage de la texture sont effectués par le codeur d'images (JPEG2000 dans notre cas). Sans aucun traitement particulier, le codeur ne prend pas en compte le fait que l'image finale à reconstruire est une version déformée de la texture décodée : il optimise la quantification de manière à reconstruire au mieux T et non pas I. En termes plus formels, pour un débit fixé, l'encodeur cherche la famille de pas de quantification $\mathcal{Q}^\star$ telle que :

$$\mathcal{Q}^\star = \arg\min_{\mathcal{Q}} \sum_{(u,v)} (T(u,v) - \hat{T}(u,v))^2 \quad \forall (u,v) \in \tilde{\mathcal{D}} \tag{4.25}$$

Or, il serait plus cohérent de déterminer la famille de pas de quantification qui minimise la distorsion de l'image synthétisée en bout de chaîne. On aimerait donc avoir :

$$\mathcal{Q}^\star = \arg\min_{\mathcal{Q}} \sum_{(x,y)} (I(x,y) - \hat{I}(x,y))^2 \quad \forall (x,y) \in \mathcal{D} \tag{4.26}$$



(a) (b) (c)

Fig. 4.11 : (a) Jacobien défini sur le domaine texture $\tilde{\mathcal{D}}$, (b) Pyramide utilisée pour pondérer les sous-bandes d'ondelettes de la texture, (c) Maillage dans $\mathcal{D}$ conduisant aux valeurs du jacobien.

En raisonnant sur des signaux continus et en effectuant le changement de variable $(x, y) = w(u, v)$, il vient :

$$\begin{aligned} \mathcal{Q}^\star &= \arg\min_{\mathcal{Q}} \int\int_{\mathcal{D}} (I(x,y) - \hat{I}(x,y))^2 \mathrm{d}x\mathrm{d}y \\ \overset{(x,y)=w(u,v)}{\Leftrightarrow} \quad \mathcal{Q}^\star &= \arg\min_{\mathcal{Q}} \int\int_{\tilde{\mathcal{D}}} (I(w(u,v)) - \hat{I}(w(u,v)))^2 J_w(u,v) \mathrm{d}u\mathrm{d}v \\ \mathcal{Q}^\star &= \arg\min_{\mathcal{Q}} \int\int_{\tilde{\mathcal{D}}} (T(u,v)) - \hat{T}(u,v))^2 J_w(u,v) \mathrm{d}u\mathrm{d}v \end{aligned} \tag{4.27}$$

On comprend donc que pour obtenir la famille de pas de quantification désirée, il faut pondérer la texture avec la racine du jacobien $\sqrt{J_w}$ avant de l'envoyer à l'encodeur. Cette pondération revient à adapter le pas de quantification aux déformations locales subies par l'image. On peut interpréter cette quantification adaptative de la façon suivante :

- Une perte de résolution est observée lorsque la superficie d'une maille diminue lors du passage de $\mathcal{D}$ à $\tilde{\mathcal{D}}$, ce qui équivaut à un jacobien supérieur à 1. Puisque le nombre d'échantillons de texture dans ces régions est plus petit que le nombre de pixels à synthétiser dans le domaine image, il semble cohérent de coder ces échantillons avec une précision plus grande pour limiter la distorsion dans les hauts-débits,
- A contrario, un gain de résolution est observé lorsque la superficie d'une maille augmente lors du passage de $\mathcal{D}$ à $\tilde{\mathcal{D}}$, ce qui équivaut à un jacobien inférieur à 1. Puisque le nombre d'échantillons de texture dans ces régions est plus grand que le nombre de pixels à synthétiser dans le domaine image, il semble cohérent d'autoriser une plus grande distorsion dans ces zones afin de réduire le débit pour un même niveau de qualité.

La figure 4.11(a) montre les valeurs du jacobien J_w en chaque pixel (u, v) du domaine texture. Un niveau de gris égal à 128 correspond à un jacobien égal à 1. Le maillage dans $\mathcal{D}$ conduisant à ce jacobien est affiché figure 4.11(c). On voit que les niveaux de gris

très inférieurs à 128 correspondent à des mailles qui se sont contractées dans le domaine image ($J_w \ll 1$). De même, les niveaux de gris très supérieurs à 128 correspondent à des mailles qui se sont étirées dans le domaine image ($J_w \gg 1$). On remarque également que le jacobien est discontinu aux frontières des mailles. Ceci est dû au fait que la transformation bilinéaire est définie indépendamment sur chaque maille. Du fait de ces discontinuités, pondérer la texture directement avec les valeurs du jacobien n'est pas judicieux car cela conduit à ajouter des hautes fréquences dans la texture et ainsi augmenter son coût de codage.

Nous proposons donc d'effectuer la pondération dans le domaine ondelettes. La valeur du jacobien est tout d'abord calculée en chaque pixel de $\tilde{\mathcal{D}}$. Ensuite, la décomposition en ondelettes de la texture sur J niveaux est générée. Pour chaque échelle $j \in \{1 \dots J\}$, chacune des trois sous-bandes de détails est alors pondérée avec une même version décimée du jacobien. Comme le montre la figure 4.12, le poids associé à chaque coefficient d'ondelette est choisi comme la valeur maximale (« MAX ») du jacobien dans une fenêtre centrée autour de la position de ce coefficient dans le domaine spatial d'origine. La taille de la fenêtre augmente avec l'échelle de la sous-bande. Le « MAX » est choisi comme heuristique pour être certain de ne pas affecter un poids faible à des zones ayant subies une perte de résolution. La figure 4.11(b) montre les poids affectés aux sous-bandes aux échelles $j = \{1, 2, 3\}$. Après avoir pondéré les coefficients dans le domaine ondelettes, une décomposition en ondelettes inverse est réalisée pour reconstruire une texture qui est envoyée au codeur JPEG2000 comme dans le schéma de base. Nous verrons au paragraphe 4.3.2 les résultats de la quantification adaptative par rapport à un codage direct de la texture.

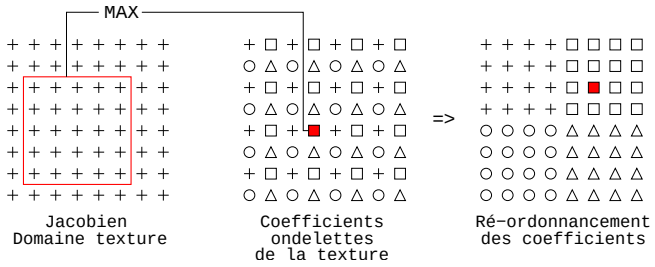


Fig. 4.12 : Pondération des coefficients d'ondelettes de la texture à partir du jacobien. Le poids associé à un coefficient d'ondelettes est le maximum du jacobien dans une fenêtre centrée sur la position du coefficient dans le domaine spatial d'origine.

4.3.1.3 Codage du maillage

Les paramètres de déformation à transmettre sont la longueur l_a d'une arête dans $\tilde{\mathcal{D}}$ et les positions des N_s sommets *internes* $\{\mathbf{x}_i = (x_i, y_i) \quad \backslash \quad i = 1 \dots N_s\}$ du maillage $\mathcal{M}$ dans $\mathcal{D}$. En pratique, il est préférable d'encoder les déplacements $\Delta\mathbf{x}_i = \mathbf{x}_i - \mathbf{u}_i$ car leurs composantes ont une énergie plus faible que les coordonnées des sommets. Les déplacements peuvent être quantifiés sans transformation préalable. Dans la suite, nous noterons Q_g le pas de quantification appliqué aux déplacements dans le domaine spatial

(pas de quantification de la géométrie). Après quantification, nous proposons de coder les symboles en plans de bits à l'aide d'un codeur arithmétique. Ceci permet de générer un flux scalable en qualité. Il est important de souligner que *la texture à encoder est calculée avec le maillage quantifié de plus haute qualité.*

Comme les déplacements sont définis sur une grille carrée (le maillage $\tilde{\mathcal{M}}$), il est possible de les décomposer dans une base d'ondelettes avant d'effectuer la quantification. Cependant, comme l'illustre la figure 4.13, effectuer la quantification dans le domaine ondelettes n'apporte pas de gain par rapport à la première technique qui est par ailleurs plus simple. Dans la suite nous effectuerons donc la quantification dans le domaine spatial. La part de débit occupée par le maillage (la géométrie) sera notée $\mathbf{R}_g$.

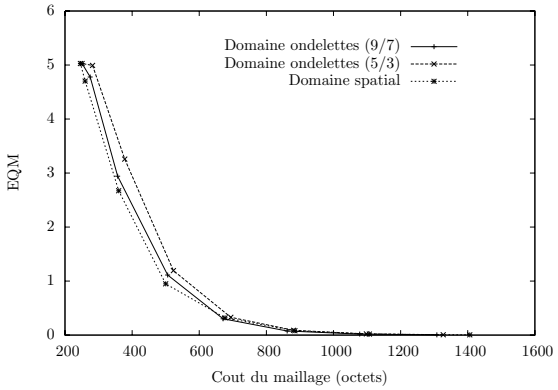


Fig. 4.13 : Courbes débit-distorsion *du maillage* de la figure 4.9 obtenues en le quantifiant dans le domaine spatial et dans le domaine ondelettes avec différents pas. Les ondelettes 9/7 et 5/3 ont été testées mais n'apportent pas de gain par rapport à une quantification dans le domaine spatial.

4.3.2 Influence des paramètres

Les résultats du schéma proposé en termes de compression peuvent varier en fonction des paramètres d'analyse $\{J, l_a, \omega_d\}$, du choix de quantifier la texture de façon adaptative ou non, et du pas de quantification Q_g utilisé pour le maillage. Toutes les méthodes utilisant un modèle de géométrie doivent trouver un compromis juste entre une adaptativité forte au contenu de l'image et un faible coût des paramètres du modèle. Comme nous l'avons décrit au chapitre 2, pour trouver le jeu de paramètres optimal parmi un ensemble de candidats, beaucoup de méthodes antérieures [PM05, Cha05b, Vel05b] basées blocs testent les candidats de façon exhaustive à l'intérieur de chaque bloc indépendamment de ses voisins. Dans notre cas, une telle méthode peut difficilement être mise en place car elle nécessiterait de répéter l'analyse pour chaque candidat possible du jeu de paramètres $\{J, l_a, \omega_d\}$, ce qui serait trop lourd en temps de calcul. Pour le pas de quantification Q_g, une méthode exhaustive pourrait être mise en place en simulant le codage-décodage et la synthèse pour chaque pas possible. Ci-dessous, nous montrons l'influence de chaque paramètre pris indépendamment, en commençant par les paramètres d'analyse. L'image considérée est *Lena* 256×256. Dans tous les résultats

présentés, le débit donné prend en compte à la fois le débit occupé par la texture $\mathbf{R}_T$ et celui occupé par le maillage $\mathbf{R}_g$. La qualité de l'image reconstruite en bout de chaîne est mesurée par le PSNR.

4.3.2.1 Influence des paramètres d'analyse J et ω_d

La figure 4.14(a) compare les points débit-distorsion obtenus en modifiant le niveau de décomposition J. Les autres paramètres sont $\{l_a = 8, \omega_d = 0.0, Q_g = 1.0\}$. Comme on peut l'observer, le niveau de décomposition a peu d'influence sur les points débit-distorsion. La figure 4.15 montre cependant les maillages obtenus en sortie d'analyse pour $J = \{1, 3, 6\}$. Nous voyons que prendre en compte trois sous-bandes de hautes fréquences dans le coût de description permet de capturer davantage de contours et de façon plus significative. Grâce à cela, nous avons constaté visuellement une meilleure reconstruction des contours. Dans la suite de nos travaux, nous utiliserons le paramètre $J = 4$ car, au-delà, peu de différences sont notables à l'œil nu.

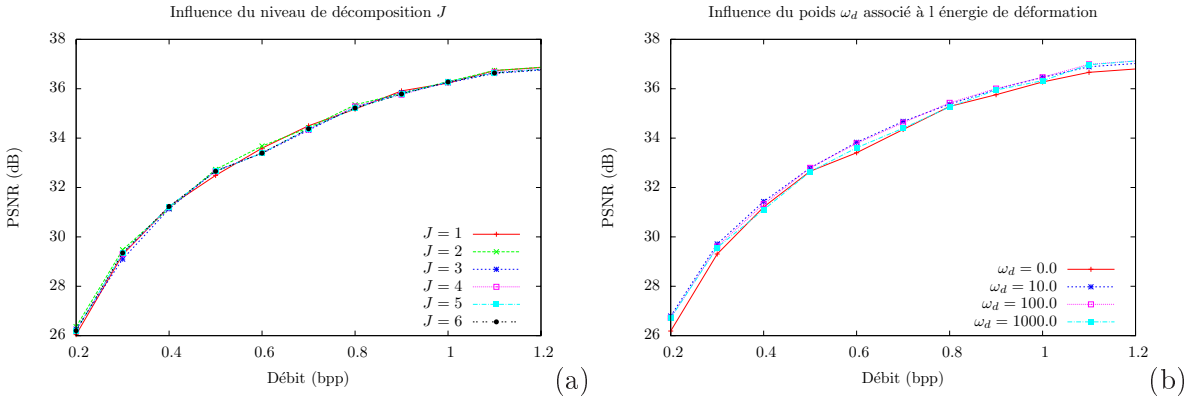


Fig. 4.14 : Influence du niveau de décomposion J et du poids ω_d associé à l'énergie de déformation $\mathbf{E}_d$.

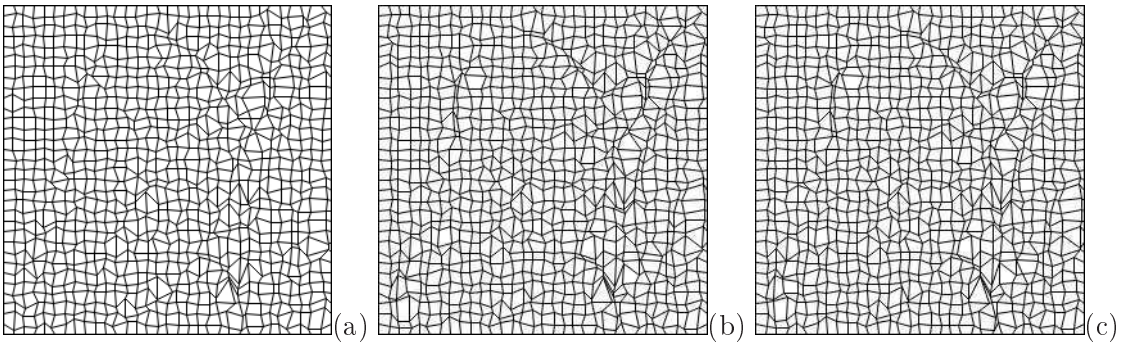


Fig. 4.15 : Influence du niveau de décomposion J. (a) $J = 1$, $\mathbf{R}_g = 0,094$ bpp, (b) $J = 3$, $\mathbf{R}_g = 0,099$ bpp, (c) $J = 6$, $\mathbf{R}_g = 0,099$ bpp.

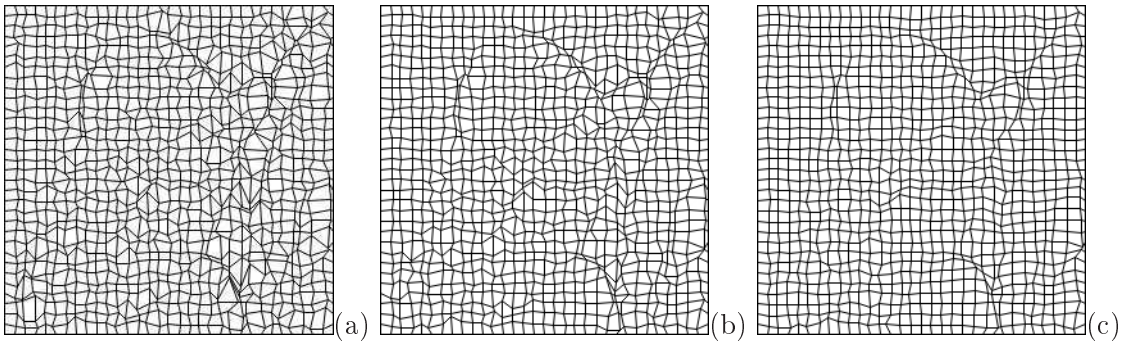


Fig. 4.16 : Influence du poids associé à l'énergie de déformation. (a) $\omega_d = 0.0$, $\mathbf{R}_g = 0,099$ bpp, (b) $\omega_d = 10.0$, $\mathbf{R}_g = 0,088$ bpp, (c) $\omega_d = 100.0$, $\mathbf{R}_g = 0,081$ bpp.

La figure 4.14(b) montre de même l'influence du poids ω_d associé à l'énergie de déformation sur les points débit-distorsion. Les autres paramètres sont $\{J = 4, l_a = 8, Q_g = 1.0\}$. La figure 4.16 montre les maillages obtenus avec les poids $\omega_d = \{0.0, 10.0, 100.0\}$. Comme on s'y attend, augmenter le poids a pour effet de « lisser » les positions. L'adaptation à la géométrie de l'image est alors moins fine mais le débit économisé peut permettre de ré-hausser la qualité des zones texturées. Ceci explique l'augmentation du PSNR en passant de $\omega_d = 0.0$ à $\omega_d = 10.0$. Nous remarquons cependant que pour les poids $\omega_d = 100.0$ et $\omega_d = 1000.0$, de légères baisses dans les résultats numériques sont observés. Ceci montre qu'il existe donc bien un compromis entre le gain en débit gagné en déformant l'image et le coût de cette déformation.

4.3.2.2 Influence de la quantification adaptative de la texture

Dans ce paragraphe, nous évaluons l'efficacité de la quantification adaptative de la texture décrite page 135. Après avoir effectué l'analyse avec le jeu de paramètres $\{J = 4, \omega_d = 10.0, l_a = 8, Q_g = 1.0\}$, nous réalisons la quantification adaptative dans le domaine ondelettes puis reconstruisons la texture avec les coefficients d'ondelettes pondérés. Cette texture modifiée est ensuite envoyée à JPEG2000. La figure 4.17 montre les courbes débit-distorsion obtenues avec et sans quantification adaptative. Nous remarquons que la quantification adaptative n'améliore pas les résultats numériques. Nos observations visuelles sont en accord avec ce constat, même s'il est parfois difficile de différencier les images reconstruites en bout de chaîne. Il y a donc un écart entre nos hypothèses théoriques et les observations pratiques. Deux explications peuvent être données. D'une part, nous remarquons que les pertes de résolution les plus significatives ont lieu sur des zones homogènes proches de contours : en se rapprochant des contours, les nœuds provoquent un étirement des régions avoisinantes (voir par exemple la figure 4.16). Ainsi, au cours de la quantification adaptative ces régions se voient affectées un poids fort alors qu'elles n'ont pas de fort impact visuel. D'autre part, même si la pondération est effectuée dans le domaine ondelettes, elle reste discontinue dans chaque sous-bande comme le montre la pyramide de poids utilisée pour les sous-bandes des trois premières échelles et illustrées figure 4.11. Du fait des discontinuités, il est possible que

l'encodage de la texture réalisé par JPEG2000 soit moins efficace. Dans la suite, nous décidons donc de ne pas activer la quantification adaptative.

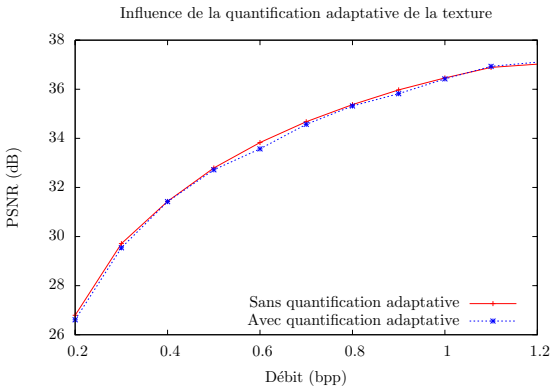


Fig. 4.17 : Influence de la quantification adaptative de la texture.

4.3.2.3 Influence du pas de quantification Q_g et de la taille l_a d'une arête

Dans ce paragraphe, nous étudions tout d'abord l'influence du pas de quantification Q_g des positions du maillage en prenant une taille d'arête $l_a = 8$. La figure 4.18(a) montre les courbes obtenues en déplaçant Q_g dans l'intervalle $[0,25; 16]$. Nous remarquons que plus le pas est grand meilleurs sont les résultats. Or, avec un pas $Q_g = 16$ la déformation opérée est l'identité. Ce résultat suggère qu'un maillage avec une taille de maille $l_a = 8$ coûte trop cher à coder : quelque soit le pas de quantification, le débit gagné en déformant l'image ne compense pas le débit pris par le maillage. Pour mieux se rendre compte de ce qui est gagné et perdu, nous affichons sur la figure 4.19 les images reconstruites à $0,4$ bpp avec les pas $Q_g = 0,25$ et $Q_g = 16$, ainsi que les images de résidus obtenues en calculant l'erreur absolue de reconstruction par rapport à l'image *Lena* d'origine. En examinant les images reconstruites, nous voyons qu'un petit pas de quantification ($Q_g = 0,25$) permet de reconstruire des contours plus nets qu'un grand pas ($Q_g = 16$) où les rebonds des ondelettes apparaissent. Les images d'erreur démontrent la diminution du phénomène de rebonds au voisinage de nombreux contours comme les contours de l'épaule ou du chapeau. En contrepartie de cette adaptation aux contours, certains détails sont perdus, notamment dans les régions texturées comme le ruban du chapeau ou les plumes. Notons qu'à ce débit, la qualité générale des deux images reste très comparable. A des débits plus faibles, la qualité visuelle de l'image reconstruite avec des pas $Q_g = 0.25, 0.5, 1.0, 2.0$ est moins bonne qu'avec un pas $Q_g = 16$ à cause du coût du maillage. Au-delà d'un pas $Q_g = 2.0$, nous pensons que la modélisation géométrique n'a plus de sens.

Pour rechercher un meilleur compromis, nous augmentons la taille des mailles. Sur la figure 4.18(b), nous avons représenté l'influence du pas de quantification en considérant cette fois une taille de maille $l_a = 16$. Comme on le voit sur cette figure, le pas de quantification $Q_g = 16$ n'est plus celui qui donne les meilleures performances. En

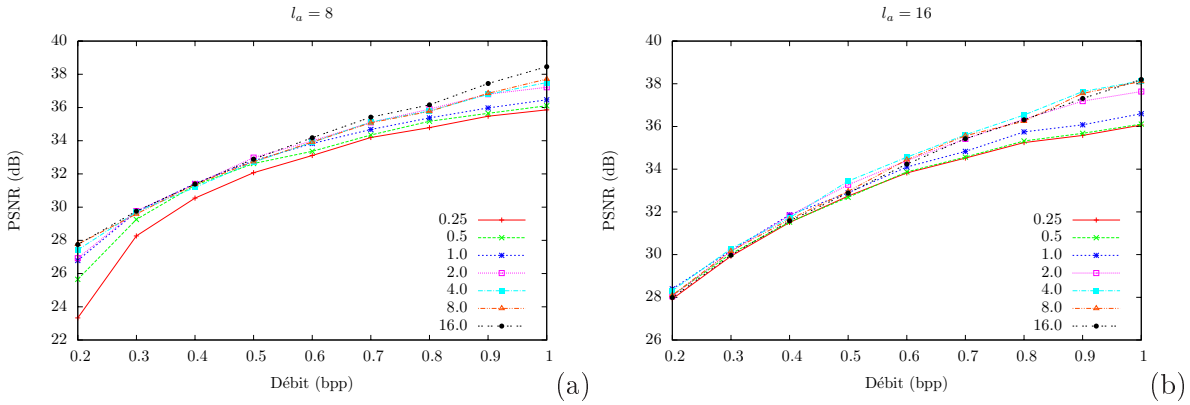


Fig. 4.18 : Influence du pas de quantification. (a) $l_a = 8$, (b) $l_a = 16$.

particulier, jusqu'à un débit de $0,5$ bpp, les pas $\{1, 2, 4\}$ donnent des PSNR légèrement meilleurs. Au delà de $0,5$ bpp, les performances avec les pas $\{0.25, 0.5, 1.0\}$ sont sensiblement moins bonnes qu'avec les pas plus élevés. L'explication de ce phénomène est la suivante. En utilisant un pas de quantification élevé, on se rapproche de la déformation identité et on limite donc les pertes dues au ré-échantillonnage. Au décodage, en se déplaçant vers les hauts débits, ceci permet de continuer à reconstruire certains détails des textures de l'image qui n'ont pas été transmis avec un pas plus faible. Ces remarques seront confirmées par les résultats visuels du paragraphe suivant.

4.3.3 Premières comparaisons avec JPEG2000

Dans ce paragraphe, nous comparons la méthode proposée avec le standard JPEG2000. Les résultats de compression pour JPEG2000 ont été obtenus en utilisant le VM 8.0 en utilisant les mêmes paramètres que pour l'encodage de la texture (voir page 135). Pour notre méthode, nous avons vu plus haut que choisir une taille d'arête $l_a = 8$ conduit à un maillage trop cher à coder. Dans les comparaisons suivantes, nous choisissons donc une taille $l_a = 16$. Avec ce choix, l'adaptation à des motifs géométriques fins ou peu espacés est limitée. Le schéma ne peut alors apporter de gain sur des images où les zones texturées dominent comme l'image test *Mandrill*, ou bien sur des images combinant des objets géométriques et des textures complexes, comme l'image test *Barbara*. Pour nos premières comparaisons avec JPEG2000, nous utilisons donc les images *Lena* et *Cameraman* qui permettent de bien mettre en compétition les atouts et limites de la méthode. La dernière section de ce chapitre proposera de modifier le modèle géométrique afin de trouver un meilleur compromis pour les images possédant un contenu plus complexe.

4.3.3.1 Décodage du maillage sans perte

Les résultats présentés ici ont été obtenus en décodant le maillage sans perte : le maillage décodé est celui qui a été utilisé à l'encodage pour calculer la texture. Les paramètres utilisés pour générer ces résultats sont $\{J = 4, l_a = 16, \omega_d = 0.0, Q_g = 1.0\}$.

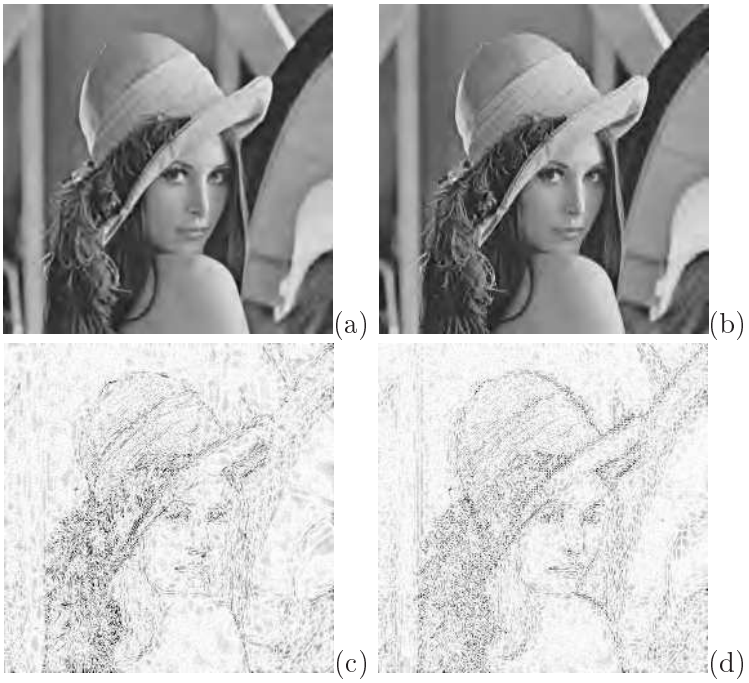


Fig. 4.19 : Influence du pas de quantification. Résultat visuel à $0,4$ bpp avec $l_a = 8$. (a) Image reconstruite avec $Q_g = 0.25$, $\mathbf{R}_g = 0,150$ bpp, PSNR=$30,54$ dB et (c) Image d'erreur magnifiée par 5. (b) Image reconstruite avec $Q_g = 16$, $\mathbf{R}_g = 0,03$ bpp, PSNR=$31,39$ dB et (d) Image d'erreur magnifiée par 5.

Les points débit-distorsion ont été obtenus en décodant un même flux pour différents débits. La figure 4.20 montre les courbes débit-distorsion obtenues pour *Lena* et *Cameraman* avec JPEG2000 et notre schéma noté « AS2D ». Deux mesures ont été utilisées pour évaluer la qualité des images reconstruites : le PSNR et la métrique SSIM proposée par Wang et al. [WBSS04] qui prend en compte certaines caractéristiques du système visuel humain. Précision que le SSIM évolue entre 0.0 et 1.0 où 1.0 signifie que l'image reconstruite est l'image d'origine.

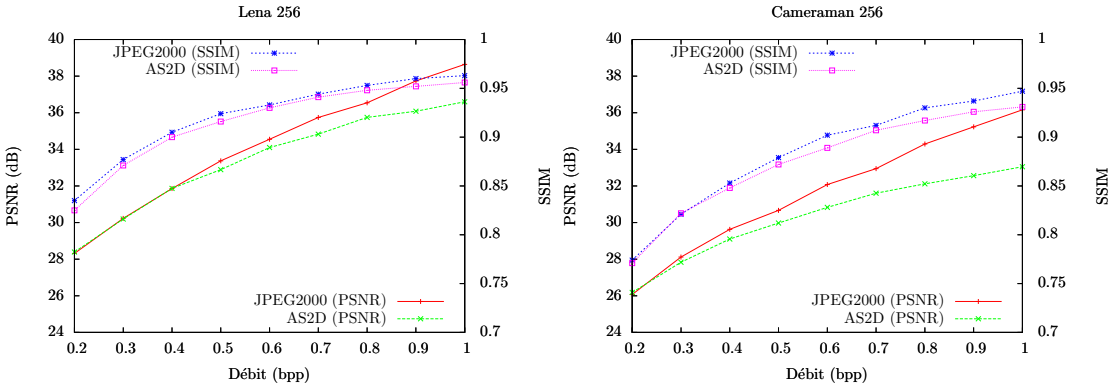


Fig. 4.20 : Comparaisons entre JPEG2000 et le schéma « AS2D »proposé.

Au regard des courbes, on remarque que les résultats numériques obtenus avec notre méthode sont globalement moins bons que ceux obtenus avec JPEG2000. Pour les débits inférieurs à 0.4 bpp, notre méthode fournit des performances numériques équivalentes à celles de JPEG2000, au niveau PSNR pour *Lena* et au niveau SSIM pour *Cameraman*. La figure 4.21 montre les images reconstruites à $0,3$ bpp avec les deux techniques. Avec le schéma proposé, on observe que certains contours sont reconstruits de façon plus nette : c'est le cas par exemple des contours de l'épaule et des contours du manteau et de la tour dans *Cameraman*. Un léger flou s'est introduit comparé à JPEG2000 qui est peu gênant à ce débit. Nous estimons par ailleurs que la qualité visuelle générale des images illustrées est meilleure avec le schéma AS2D.

Lorsqu'on se déplace dans les hauts débits, l'écart de PSNR augmente sensiblement du fait des pertes dues aux ré-échantillonnages. La figure 4.22 montre sur la colonne de gauche l'erreur absolue de reconstruction (multipliée par 5) à $0,9$ bpp dans le cas de JPEG2000 et du schéma AS2D. La colonne de droite montre les valeurs du SSIM. Examinons tout d'abord l'erreur absolue de reconstruction. Dans le cas du schéma AS2D, on remarque que cette erreur est globalement plus importante que celle obtenue avec JPEG2000. Nous distinguons ici l'erreur commise sur les zones texturées et l'erreur commise sur les contours. Dans le cas des zones texturées, l'erreur a un impact sur la qualité visuelle des images reconstruites : nous avons par exemple constaté un flou sur les plumes de *Lena*. Cette observation est supportée par l'index SSIM. On voit en effet qu'au niveau des plumes du chapeau le SSIM est plus faible dans le cas de notre schéma que dans le cas de JPEG2000. Dans le cas des contours, l'erreur n'a pas d'impact sur la qualité visuelle. En effet, au niveau des contours comme ceux du chapeau ou de

Fig. 4.21 : Comparaisons entre les images reconstruites à 0, 3 bpp avec JPEG2000 à gauche et le schéma AS2D proposé à droite.

l'épaule, les déformations effectuées ont introduit un *gain* de résolution. De ce fait, l'erreur commise du fait de l'aller-retour entre le domaine image et le domaine texture n'est pas visible à l'œil nu. Il en va de même pour JPEG2000 *à ce débit* et l'index SSIM corrobore ses observations car il est très proche de 1 au niveau des contours. Introduire un léger bruit près des contours n'a donc pas d'impact visuel.

4.3.3.2 Décodage du maillage avec pertes

Dans [Cam04b], l'auteur décrit un schéma de codage vidéo par analyse-synthèse où le *mouvement* est modélisé par un maillage déformable. Comme dans notre technique, les déformations appliquées aux images sont continues sur tout le domaine. Au décodage, l'auteur observe alors qu'une légère perte sur l'information de mouvement a peu d'impact sur la qualité *visuelle* du mouvement reconstruit. En outre, cette perte sur le mouvement permet de reporter une portion du débit sur le décodage des textures. L'auteur observe que ceci a pour effet de rehausser la qualité *visuelle* générale des images reconstruites et conclut donc que *l'œil humain est plus sensible à une perte sur les textures qu'à une perte sur le mouvement*. Bien sûr, comme le mouvement est reconstruit avec perte, certaines caractéristiques des images se trouvent décalées par rapport à leur position d'origine et ceci réduit radicalement la mesure du PSNR (même lorsque le décalage est à l'échelle sous-pixellique).

Dans nos travaux, nous nous sommes de même intéressés à l'impact visuel que peut avoir une perte *géométrique* sur la qualité de l'image reconstruite en bout de

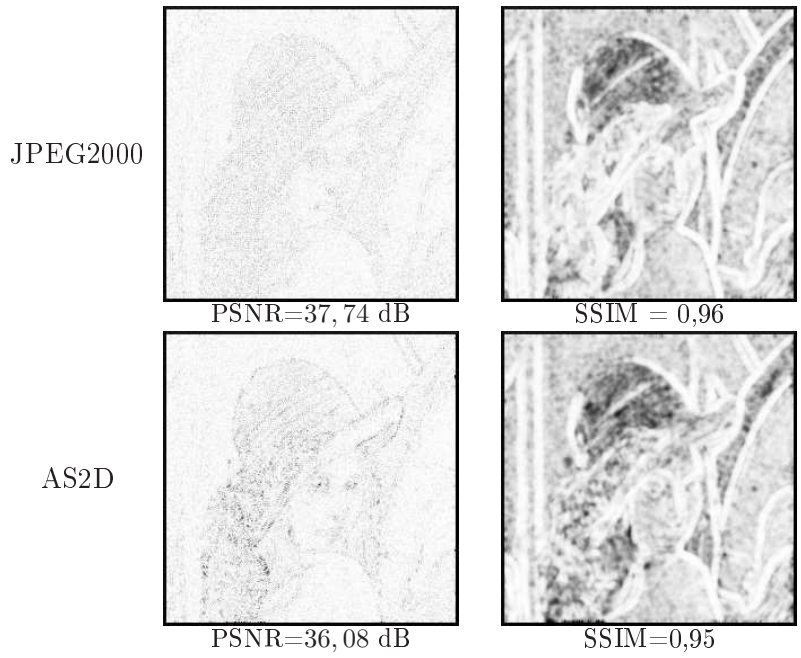


Fig. 4.22 : Erreur absolue et index SSIM à 0.9 bpp pour JPEG2000 et le schéma AS2D. Pour le SSIM, plus le niveau de gris est élevé (zones claires), plus la qualité est proche de celle de l'image d'origine.

chaîne. Travaillons ici sur l'image *Cameraman*. Nous encodons le maillage avec un pas de quantification $Q_g = 0,25$. Comme nous l'avons vu plus haut, le maillage est encodé en plans de bits pour générer un flux scalable. Au décodage, il est donc possible de tronquer le flux en ne décodant pas les derniers plans de bits. Soit n_p le nombre de plans de bits non décodés à reception. La figure 4.23 montre les images reconstruites à $0,3$ bpp en prenant $n_p = \{0, 2, 3\}$. $n_p = 0$ correspond à un décodage sans perte du maillage. La première observation que nous pouvons faire est que la qualité des images reconstruites avec $n_p = 0$ et $n_p = 2$ est très similaire, même si on note une diminution du PSNR d'environ 2 dB. On voit donc qu'une légère perte sur la déformation a peu d'impact sur la qualité *visuelle* de l'image reconstruite. En notant $\hat{w}$ le maillage décodé, cette observation peut se traduire mathématiquement par :

$$I(((\hat{w})^{-1} \circ w)(x, y)) \overset{visu}{\approx} I(x, y) \quad \forall (x, y) \in \mathcal{D} \tag{4.28}$$

La seconde observation que nous pouvons faire est que libérer une part du débit en décodant le maillage avec perte n'apporte pas de gain *visuel* significatif sur les zones texturées, pour le cas traité ici. L'explication est que le coût du maillage encodé (ici $0,04$ bpp) reste marginal par rapport au coût de la texture et donc un gain de l'ordre de $0,01$ bpp pour une image de taille 256×256 ne se traduit pas par un gain visuel. On peut supposer qu'un tel gain pour une image au format *SD* aurait un impact visuel positif. Pour évaluer le gain sur la texture apporté par un décodage du maillage avec perte, nous proposons d'observer le PSNR de la texture décodée par rapport à la texture

$n_p = 0$, $PSNR = 27,42$ dB $n_p = 2$, $PSNR = 25,15$ dB $n_p = 3$, $PSNR = 24,57$ dB

Fig. 4.23 : Image reconstruite à $0,3$ bpp en tronquant n_p plans de bits de la géométrie. La part de débit prise par le maillage est $0,04$ bpp pour $n_p = 0$, $0,022$ bpp pour $n_p = 2$ et $0,017$ bpp pour $n_p = 3$.

issue de l'analyse. Cette mesure est affichée sur la figure 4.24. Sur cette figure, le débit prend en compte à la fois le débit de la texture et du maillage décodés. D'une façon générale, on constate une amélioration du PSNR texture au fur et mesure que l'on perd de l'information sur la géométrie. Ce gain est plus net dans les bas débits, mais n'est donc pas encore suffisamment significatif pour se traduire visuellement.

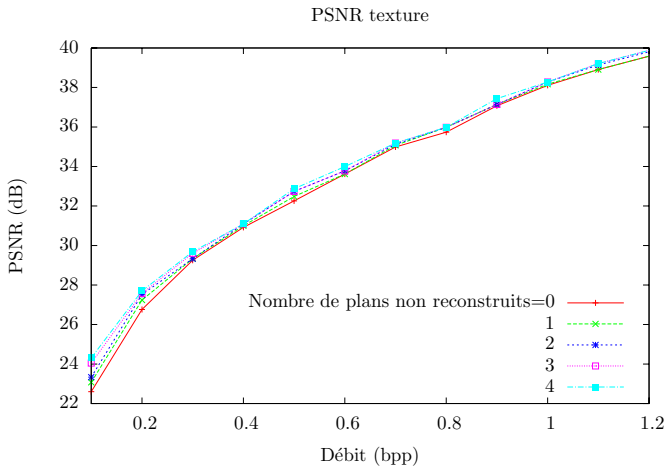


Fig. 4.24 : PSNR de la texture reconstruite en libérant progressivement la bande passante prise par l'information de déformation.

Sur la dernière colonne de la figure 4.23, nous montrons l'image reconstruite en tronquant 3 plans de bits. Dans ce cas, on peut observer un léger gain au niveau de l'herbe. Cependant, dans ce cas la perte sur la géométrie est visible, particulièrement au niveau du bras et de l'épaule. Insistons cependant sur le fait que la scalabilité géométrique peut aller de pair avec une scalabilité spatiale. Plus précisément, si l'on reconstruit l'image à une résolution moindre que sa résolution d'origine, il semble logique de réduire la précision du modèle géométrique en conséquence. Ainsi, si une image de dimension 256×256 est encodée avec un maillage de précision Q_g puis décodée à une résolution

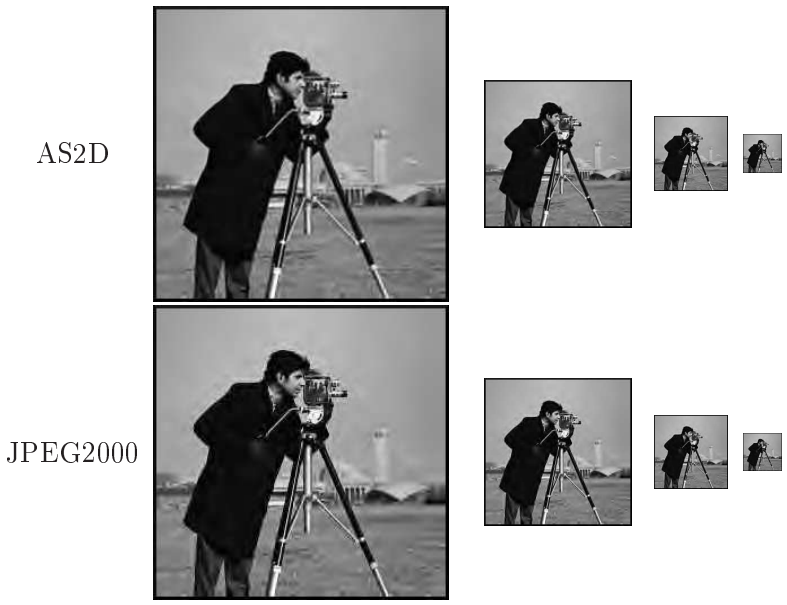


Fig. 4.25 : Image reconstruite à $0,3$ bpp en tronquant n_p plans de bits de la géométrie.

spatiale 128×128, il paraît logique de décoder le maillage avec un précision $Q_g/2$. La figure 4.25 montre ainsi deux pyramides multi-résolutions : celle du bas représente l'image reconstruite à $0,3$ bpp avec JPEG2000 ainsi que ses versions décimées d'un facteur $\{2,4,6\}$; celle du haut représente l'image reconstruite à $0,3$ bpp avec le schéma AS2D puis les versions obtenues en tronquant $n_p = \{1,2,3\}$ plans de bits et en décimant l'image synthétisée d'un facteur $\{2,4,6\}$. Nous remarquons qu'en adaptant la résolution de l'image aux pertes sur la géométrie, ces pertes ne sont guère détectables à l'œil nu et l'équation (4.28) est satisfaite de façon plus générale.

4.3.4 Premier bilan

Dans cette section, nous avons présenté les résultats de compression que nous avons obtenus avec le schéma par analyse-synthèse. En particulier, nous avons vu que le choix des différents paramètres influence le compromis entre une adaptativité forte du maillage au contenu de l'image et un coût faible de l'information géométrique. Ce compromis dicte le compromis débit-distorsion obtenu en bout de chaîne. Nous mettons en avant deux limites du schéma qui motivent les travaux de la section suivante :

Coût de la géométrie. Au paragraphe 4.3.2.3, nous avons conclu qu'un maillage avec une taille de maille de l'ordre de 8×8 coûte trop cher à coder. Nous avons alors présenté des résultats en considérant une taille de maille de l'ordre de 16×16. Des gains ont été observés au niveau des contours sur des images comme *Lena* et *Cameraman* contenant une géométrie peu complexe. Pour s'adapter à des géométries plus complexes, une taille de maille inférieure est nécessaire.

Reconstruction des textures. En comparant notre technique avec JPEG2000, nous avons observé un gain visuel dans les bas débits mais nous avons aussi constaté une perte au niveau des zones texturées dans les hauts débits. Cette perte est due aux ré-échantillonnages effectués lors de l'aller-retour du domaine image au domaine texture. Si on souhaite élargir le champ d'applications à une gamme plus large de débits, il est important de modifier le schéma en prenant cela en compte.

Dans la section suivante, nous proposons trois méthodes simples que nous avons testées dans le but de résoudre ces limites et ainsi améliorer le compromis débit-distorsion. Les deux premières méthodes tentent d'apporter des solutions au problème des zones texturées. Dans la première, nous proposons d'encoder et transmettre une image d'erreur en plus de la texture. Dans la seconde, nous proposons d'encoder une texture de résolution supérieure à celle de l'image d'origine afin de limiter les pertes de ré-échantillonnage. La troisième technique proposée est un post-traitement sur le maillage visant à conserver l'adaptation aux contours de l'image tout en limitant le coût des déplacements dans les zones où la déformation n'apporte pas de gain significatif. Ceci permet de rehausser la qualité des zones texturées et diminuer le coût du maillage, tout en gardant une bonne adaptation aux contours. Les résultats de ce nouveau compromis entre adaptativité et parcimonie sont étudiés.

4.4 Modifications du schéma

Les deux premières modifications au schéma que nous proposons ont pour but de rehausser la qualité visuelle des textures reconstruites dans les hauts débits. Nous reprenons les paramètres $\{J = 4, l_a = 16, \omega_d = 0.0, Q_g = 1.0\}$ utilisés pour la comparaison avec JPEG2000 et cherchons à conserver les atouts de la méthode dans les bas débits tout en relevant les courbes de PSNR dans les hauts débits.

4.4.1 Codage de l'image de résidus

A la fin de l'analyse, nous disposons d'une texture T et d'une déformation w. Avant d'envoyer la texture à JPEG2000, nous proposons ici de calculer l'image $I^\star$ en inversant la déformation. Comme nous l'avons introduit plus haut, $I^\star$ est l'image de qualité maximale qu'il est possible de reconstruire avec le schéma de base. Connaissant $I^\star$ et l'image d'origine I, nous pouvons définir une image de résidu $I_\epsilon = I - I^\star$. L'idée est alors d'encoder et transmettre l'image de résidus en plus de la texture. En pratique, nous envoyons simplement le couple (T, I_ϵ) à JPEG2000 qui génère un flux scalable à partir des deux images sans traitement spécifique de l'utilisateur. En opérant ainsi, nous ne travaillons plus à échantillonnage critique. La question est de savoir si la redondance introduite réduit les performances dans les bas débits.

Sur la figure 4.26, nous montrons les courbes débits-distorsions obtenues sur les images *Lena* et *Cameraman* avec et sans codage de l'image de résidus. La courbe donnée par JPEG2000 est aussi affichée. Nous remarquons tout d'abord que l'introduction d'une redondance modifie très peu les performances dans les bas débits. Dans les moyens et hauts débits, l'encodage de I_ϵ apporte un gain de PSNR. Ce gain se traduit au niveau

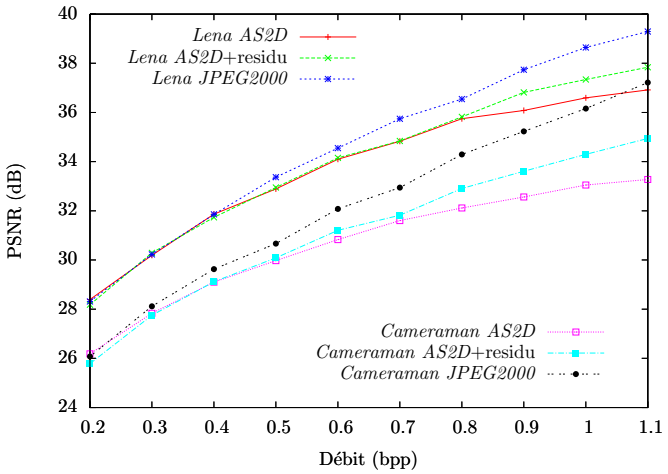


Fig. 4.26 : Encoder une image de résidus a pour effet de rehausser la valeur du PSNR dans les hauts-débits et la qualité visuelle des textures reconstruites.

visuel par une amélioration du contraste dans les textures. Il devient alors très difficile de distinguer des différences avec l'image reconstruite par JPEG2000.

4.4.2 Augmentation de la résolution de la texture

Repartons du schéma AS2D de base. Dans la description de ce schéma, aucune contrainte n'a été avancée concernant les dimensions de la texture. Aussi, rien ne force à encoder et transmettre une texture de la même dimension que l'image d'origine. Or, pour limiter les pertes dues au ré-échantillonnage dans les hauts débits, une solution simple consiste à calculer une texture ayant plus d'échantillons que l'image d'origine. Dans ce paragraphe, nous proposons donc de coder une texture dont les dimensions sont multiples de celles de l'image d'un rapport noté r_d. Le calcul de la texture requiert alors la recherche de ${r_d}^2$ fois plus de correspondants dans le domaine image. Une nouvelle fois, la question est de savoir si la redondance introduite ne détériore pas la qualité des images dans les bas débits.

La figure 4.27 répond à cette question. Dans cette figure, nous avons illustré les courbes débits-distorsions obtenues en considérant plusieurs valeurs de r_d. On remarque qu'augmenter la résolution de la texture jusqu'à $r_d = 3$ ne détériore pas les performances dans les bas débits. Ceci prouve que le contenu de la texture est bien adapté à un codage par ondelettes. Par rapport à la technique proposée précédemment, on observe une légère amélioration du PSNR dans les moyens débits. Dans les hauts débits, les remarques sont similaires à celles données au paragraphe précédent. Augmenter la résolution de la texture provoque une nette amélioration du PSNR jusqu'à $r_d = 3$. Nous avons par ailleurs constaté que $r_d = 4$ donnait des performances moins bonnes, particulièrement dans les bas débits, car le facteur de redondance devient trop élevé.

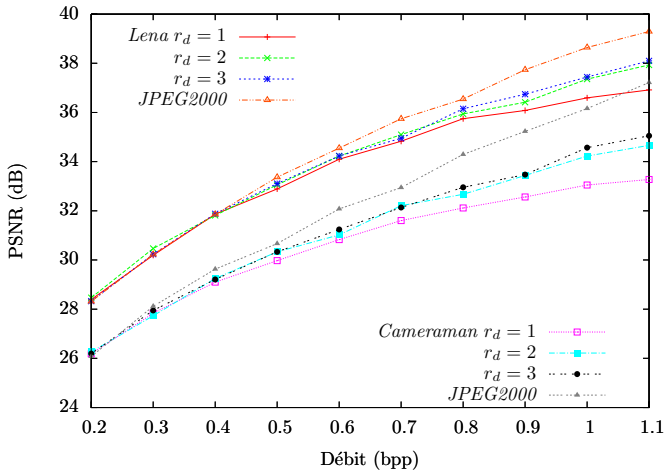


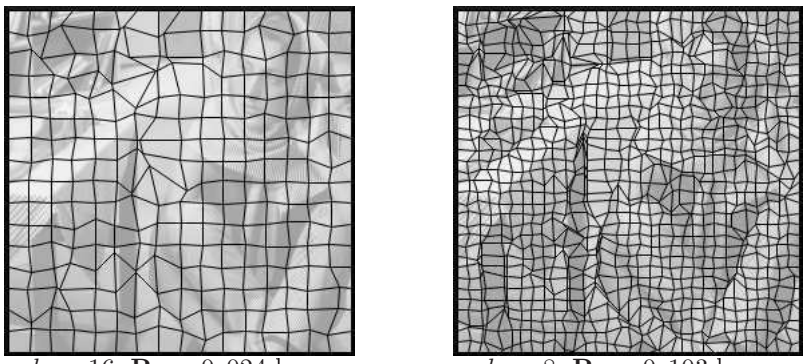
Fig. 4.27 : Effet d'une augmentation de la résolution de la texture sur les courbes débit-distorsion. r_d est le facteur de multiplication des dimensions entre l'image d'origine et la texture.

4.4.3 Amélioration du compromis adaptativité-coût

4.4.3.1 Positionnement du problème

Dans la section précédente, des tests ont montré qu'un maillage avec une taille d'arête l_a de l'ordre de 8 coûtait trop cher à coder. Ceci nous a conduit à choisir une taille $l_a = 16$ et à restreindre nos tests à des images possédant un contenu géométrique simple. Pour des images possédant un contenu un peu plus complexe, il est important d'améliorer l'adaptativité du maillage en choisissant une taille de maille plus petite. Ce constat est par exemple flagrant si l'on effectue l'analyse sur l'image *Barbara* pour $l_a = 16$ puis pour $l_a = 8$. La figure 4.28 montre les maillages obtenus dans chaque cas. Dans le cas $l_a = 16$, on voit que l'adaptation du maillage est assez limitée. Ceci s'explique par le fait que les dimensions de certains objets à modéliser ou leurs distances les uns des autres sont de l'ordre de l_a ou inférieures. Ainsi, il est difficile de capturer les contours des pieds de la table, les livres dans le fond ou encore le contour du bras de *Barbara*. Dans le cas $l_a = 8$, on remarque que ces contours sont capturés et que l'adaptation est globalement meilleure. Cependant, cette adaptation a un prix non négligeable : le coût du maillage est multiplié par 4.

En se fixant une taille d'arête $l_a = 8$, le but de cette sous-section est de chercher à réduire le coût du maillage tout en conservant une bonne adaptation aux contours. Les méthodes décrites ci-dessous sont des *post-traitements* simples à effectuer **après l'étape d'analyse**. Nous proposons tout d'abord d'annuler les déformations dans les zones texturées où l'aller-retour entre le domaine image et le domaine texture génère des pertes. « Annuler » la déformation d'une maille signifie ici la re-positionner sur le carré d'origine qu'elle occupait à l'étape (0) de l'analyse. En effectuant ceci, nous épargnons le coût des déplacements et améliorons la reconstruction des zones texturées dans les hauts débits. Pour réduire de façon plus drastique le coût du maillage, nous proposons ensuite d'annuler les déformations de toutes les mailles dont les déplacements de nœuds



$l_a = 16$, $\mathbf{R}_g = 0,024$ bpp $\quad$ $l_a = 8$, $\mathbf{R}_g = 0,103$ bpp

Fig. 4.28 : Nécessité de recourir à une taille de maille $l_a = 8$ pour *Barbara*. Les débits affichés ont été obtenus en prenant un pas de quantification $Q_g = 1.0$.

ne sont pas significatifs. En particulier, dans les zones homogènes les nœuds ont souvent tendance à se déplacer légèrement de leur position d'origine. Ces légers déplacements cumulés ont un coût non négligeable sans apporter un gain visible de qualité. Enfin, nous proposons de réduire encore le coût du maillage en représentant l'ensemble des mailles non déformées par une structure de Quadtree.

4.4.3.2 Cas des zones texturées

Penchons-nous tout d'abord sur le cas des zones texturées. Comme nous l'avons dit précédemment, les déformations dans ces zones produisent des pertes numériques. Ces pertes ont un impact sur la qualité visuelle des images après synthèse. Puisque ces déformations ont un coût et de surcroît n'apportent pas de gain visuel, la solution est de les annuler en replaçant les mailles sur leur carré d'origine. La question qui se pose ici est la suivante : comment détecter les zones texturées pour reconnaître les mailles à replacer sur leur carré d'origine ? Dans [LW95], Lee et Wang proposent d'adapter la densité des nœuds d'un maillage (utilisé dans leurs travaux comme grille d'échantillonnage) en fonction du contenu local de l'image. Pour ce faire, ils s'appuient sur des descripteurs statistiques simples [VG92] leur permettant de classer le contenu d'une maille comme étant homogène, texturé ou comme possédant un contour.

Dans le cadre de notre étude, nous proposons de tirer avantage de l'image $I^\star$ qui peut être calculée à l'issue de l'analyse en simulant l'étape de synthèse sans encodage. Cette image, par comparaison à l'originale, permet de reconnaître les régions texturées. Une première solution consiste à analyser l'image d'erreur $I_\epsilon = I - I^\star$ à l'intérieur de chaque maille et de replacer sur leur carré d'origine les mailles à l'intérieur desquelles le résidu total est supérieur à un seuil. Cette solution n'a pas été choisie pour une raison précise : l'aller-retour entre le domaine image et le domaine texture génère aussi une erreur au niveau des contours. Comme cette erreur n'est pas perceptible a l'œil nu et que les déplacements des nœuds dans ces zones apportent un gain à bas débits, nous ne souhaitons pas qu'elle ait une influence sur la géométrie finale du maillage. Plutôt que de nous appuyer sur l'erreur absolue, nous préférons donc nous baser sur

l'index SSIM. La figure 4.29 montre l'image *Barbara* d'origine I, l'image $I^\star$ obtenue à l'issue de notre analyse avec le maillage représenté figure 4.28 (cas $l_a = 8$), et enfin l'index SSIM comparant I et $I^\star$. On s'aperçoit que les valeurs les plus basses du SSIM correspondent aux zones texturées de l'image d'origine tandis que les valeurs les plus hautes correspondent aux contours.

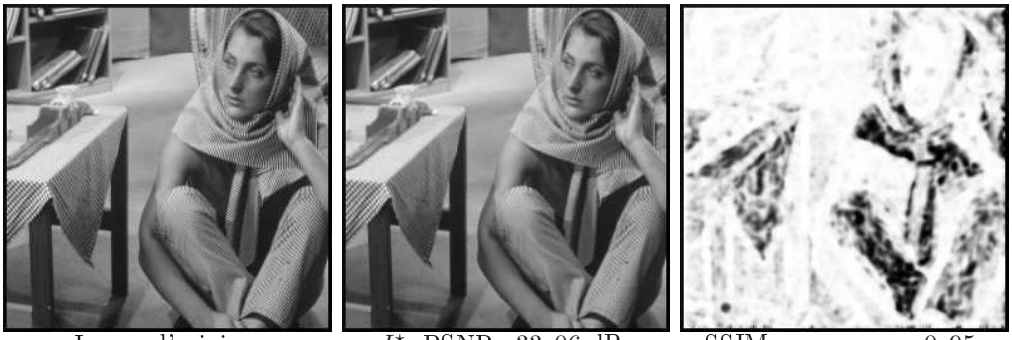


Image d'origine | $I^\star$, PSNR=33,06 dB | SSIM, moyenne = 0,95

Fig. 4.29 : Image *Barbara* d'origine et image synthétisée après un aller-retour entre le domaine image et le domaine texture avec le maillage représenté figure 4.28 (cas $l_a = 8$) .

La carte de disparité du SSIM nous permet d'effectuer un post-traitement simple pour modifier le maillage. Pour chaque maille du maillage $\mathcal{M}$ dans le domaine image, le SSIM moyen a l'intérieur de la maille est calculé. S'il est supérieur à un seuil, la forme de la maille n'est pas modifiée : ceci doit permettre de conserver une bonne adaptation aux contours. S'il est inférieur au seuil, la maille est replacée sur son carré d'origine. Le seuil est noté T_{ssim}. En pratique, il est préférable d'effectuer d'abord une boucle sur les mailles pour fixer celles qui génèrent un index SSIM élevé. Une seconde boucle permet de remettre à leur position d'origine les sommets n'appartenant pas à ces mailles fixées. Si on n'opère pas de cette façon, une maille apportant un SSIM élevé peut se trouvée modifiée si les mailles voisines qui la précèdent dans l'ordre de parcours sont replacées sur leur carré d'origine. La figure 4.30 montre à droite le résultat de la méthode en considérant un seuil $T_{ssim} = 0,95$. Nous avons reproduit à droite le maillage issu de l'analyse ainsi que la carte du SSIM comparant $I^\star$ à I. Le seuil choisi correspond à la moyenne du SSIM de $I^\star$. On remarque que les mailles replacées sur leur carré d'origine correspondent bien aux zones texturées de *Barbara*, ce qui était l'objectif de départ. On remarque que le coût du maillage ne diminue pas significativement. Cependant, on remarque visuellement que l'index SSIM de $I^\star$ dans les zones texturées a considérablement augmenté. Une augmentation du PSNR de $I^\star$ de 4,4 dB est aussi observée par rapport au schéma AS2D de base.

4.4.3.3 Déformations non significatives

Toujours dans l'optique de trouver un meilleur compromis entre adaptativité, coût du maillage et qualité de synthèse, nous observons la chose suivante : bien souvent, dans les zones homogènes, les nœuds se déplacent au cours des itérations de l'analyse. Ceci

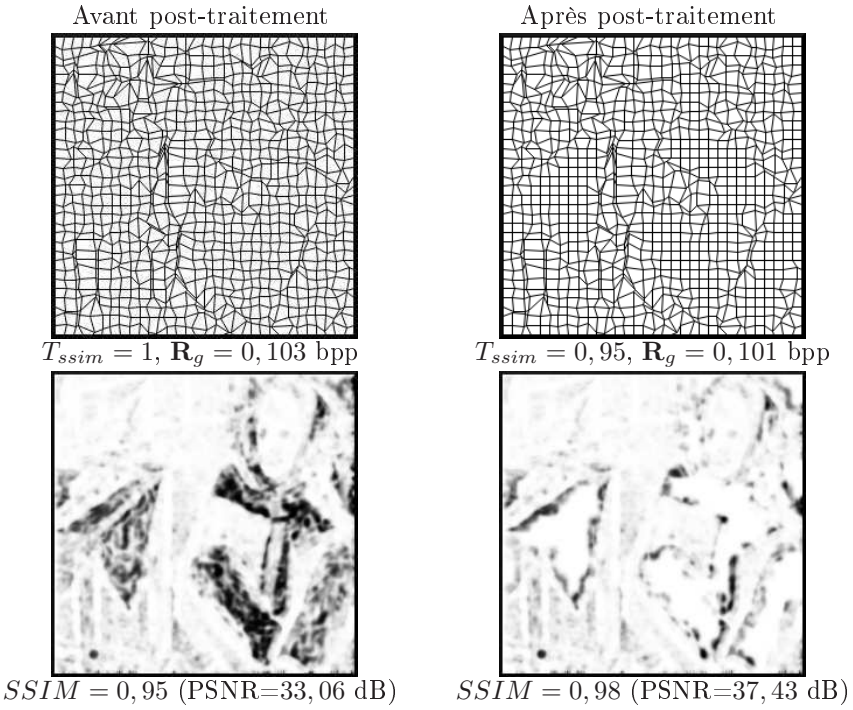
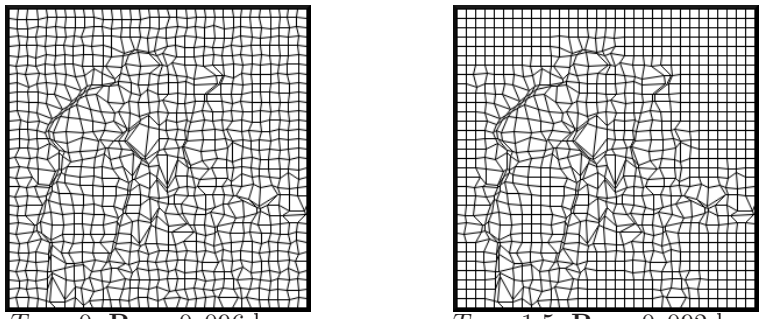


Fig. 4.30 : Post-traitement pour réduire le coût du maillage et améliorer la qualité des textures dans les hauts-débits. En haut, le maillage. En bas, index SSIM de l'image de qualité optimale qu'il est possible de reconstruire.

peut être dû à une légère activité en gradient dans ces régions, par exemple causé par un bruit d'acquisition non perçu par l'œil. Ces déplacements ont très peu d'impact (bon ou mauvais) sur la qualité de $I^\star$. Or, tous ces déplacements cumulés peuvent avoir un coût non négligeable. L'image *Barbara* a peu de surfaces homogènes de grande taille, mais d'autres images comme *Cameraman* ou des images de type cartoon présentent de grandes régions sans contours ni textures. Nous proposons donc ici un post-traitement à l'analyse pour repérer les mailles qui ont peu bougé et les replacer sur leur carré d'origine.



$T_w = 0$, $\mathbf{R}_g = 0,096$ bpp $\quad$ $T_w = 1.5$, $\mathbf{R}_g = 0,092$ bpp

Fig. 4.31 : Post-traitement pour annuler les déformations non significatives par rapport au seuil T_w.

La mesure sur laquelle nous proposons de nous appuyer pour décider si une maille a bougé significativement ou non est le jacobien de la déformation J_w. J_w est ainsi calculé en chaque point $\mathbf{u}$ du domaine texture de la même façon que lors de la quantification adaptative décrite au paragraphe 4.3.1.2. En chaque point $\mathbf{u}$, nous proposons ensuite de définir le critère de déformation $\mathbf{C}_w$ comme :

$$\mathbf{C}_w(\mathbf{u}) = \begin{cases} J_w(\mathbf{u}) & si \quad J_w(\mathbf{u}) \geqslant 1 \\ 1/J_w(\mathbf{u}) & si \quad J_w(\mathbf{u}) < 1 \end{cases} \tag{4.29}$$

$\mathbf{C}_w$ indique le changement de résolution local sans tenir compte s'il s'agit d'une perte ou d'un gain de résolution. Une fois ce critère défini, nous parcourons chaque maille 8×8 du domaine texture et calculons la moyenne de $\mathbf{C}_w$ dans chacune d'entre elles. Toutes les mailles dont le critère $\mathbf{C}_w$ moyen est inférieur à un seuil sont replacées sur leur carré d'origine [1]. Le seuil est noté T_w et vaut au minimum 1 (dans ce cas, le post-traitement n'a pas d'effet). Un traitement en deux passes comme dans la technique précédente est mis en place pour éviter que des mailles dont le critère est supérieur au seuil se trouvent modifiées par le déplacement des nœuds voisins. La figure 4.31 montre le maillage obtenu sur *Cameraman* à la fin de l'analyse et le résultat du post-traitement en prenant $T_w = 1.5$. On remarque que les déformations non significatives ont bien

[1] Remarquons qu'une valeur $\mathbf{C}_w = 1$ ne signifie pas forcément que la maille ne s'est pas déformée. On peut par exemple avoir $\mathbf{C}_w = 1$ dans le cas où le changement de résolution selon l'axe u est inversement proportionnel au changement de résolution selon l'axe v. Cependant, du fait de la contrainte de connectivité sur notre maillage, ce cas a une faible probabilité de survenir et le critère $\mathbf{C}_w$ permet bien de détecter les déformations significatives.

été détectées et le replacement des mailles sur leur carré d'origine correspond à notre attente.

Comme dans la méthode précédente, nous remarquons que le post-traitement ne réduit pas le coût du maillage de façon satisfaisante. Coder des déplacements nuls permet de gagner ici $0,004$ bpp (soit $0,004 \times 256 \times 256 \approx 260$ bits) ce qui n'est pas suffisant pour conserver les bonnes performances dans les bas débits obtenues avec une plus grande taille de maille. Cependant, nous observons que les mailles carrées sont regroupées en régions. Plutôt que de coder un grand nombre de déplacements nuls dans ces régions, nous proposons dans le paragraphe suivant de regrouper les mailles carrées dans une structure de type Quadtree.

4.4.3.4 Création d'un maillage Quadtree

Le Quadtree, ou arbre quaternaire, a été introduit au chapitre 2, page 64. Ici, nous travaillons sur le maillage $\mathcal{M}$. La racine du Quadtree correspond à l'ensemble de toutes les mailles. Une segmentation du maillage en Quadtree peut être obtenue en regroupant récursivement les mailles quatre par quatre. Chaque niveau j de l'arbre comporte 2^{j+1} nœuds correspondant à un regroupement dyadique de mailles. Pour créer un Quadtree adaptatif, il est possible d'associer à chaque nœud une valeur binaire pour décider si le nœud doit être subdivisé en quatre ou non.

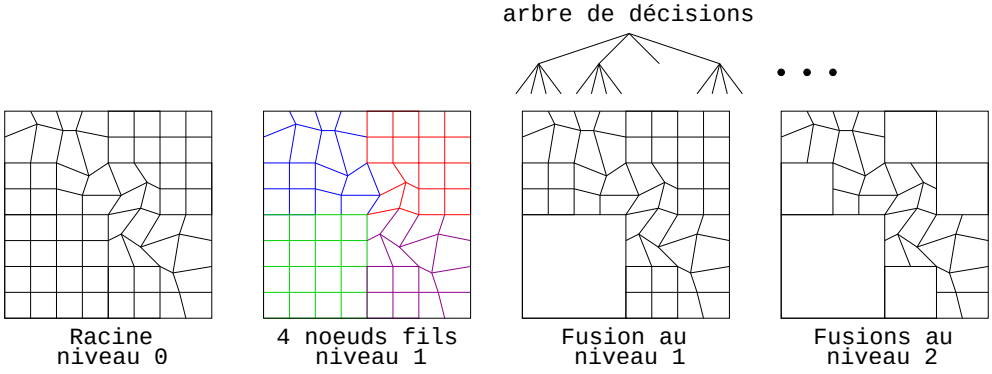


Fig. 4.32 : Fusion des mailles carrées par une approche descendante.

L'algorithme que nous proposons ici est un algorithme récursif destiné à être appliqué à l'issue des post-traitements précédents. Nous débutons à la racine du Quadtree et parcourons toutes les mailles. Si une seule d'entre elles est déformée par rapport à son carré d'origine, alors on ne peut pas les fusionner et nous associons à la racine la valeur 1. Ceci signifie qu'il faut aller au moins au niveau de profondeur supérieur pour pouvoir fusionner des mailles carrées. Si toutes les mailles sont situées sur leur carré d'origine, nous associons à la racine la valeur 0. Ceci signifie l'arrêt de l'algorithme. Si la décision est 1, l'algorithme se répète sur les quatre nœuds fils de la racine. Chacun de ces nœuds correspond à un regroupement dyadique de mailles, comme illustré figure 4.32. Au fur et à mesure des récursions, certaines branches s'arrêtent signifiant que des mailles carrées ont été regroupées au niveau de profondeur courant. D'autres branches se poursuivent

jusqu'au tout dernier niveau qui correspond à l'échelle de la maille. A ce dernier niveau, une valeur 0 est affectée à la feuille si la maille est non déformée, 1 dans le cas contraire. Si la valeur est 1, il faudra encoder les positions de la maille. Au final, le contenu de l'image est modélisé par un arbre de décisions et un ensemble de positions. Ces deux structures sont encodées séparément, en conservant un codage en plans de bits pour les positions.

La figure 4.33 montre les « maillages Quadtree » obtenus à la fin de l'analyse pour *Lena*, *Cameraman*, *Barbara* et *Peppers*. Dans chaque cas, nous donnons le débit du maillage avant et après les post-traitements, le PSNR de l'image $I^\star$ ainsi que les seuils T_{ssim} et T_w utilisés pour générer le Quadtree. Pour les trois premières images, nous remarquons une baisse significative du coût du maillage et de surcroît une hausse du PSNR de l'image $I^\star$ (pour une image de dimensions 256×256, une baisse de $0,03$ bpp correspond à un report de 245 octets sur la part de débit accordée à la texture). Pour *Peppers*, les gains sont moins significatifs car l'image contient plus d'objets et de contours (produisant des déformations de maille significatives par rapport au seuil T_w) et peu de zones texturées (donc peu de zones sous le seuil T_{ssim}). Dans le paragraphe suivant, nous présentons les résultats de compression obtenus en bout de chaîne et les comparons à ceux fournis par JPEG2000.

4.4.3.5 Résultats de compression

Les résultats de compression présentés dans ce paragraphe ont été obtenus en effectuant l'analyse, les post-traitements et l'encodage sur les quatre images tests utilisées précédemment. Pour les post-traitements, les seuils utilisés sont ceux indiqués sur la figure 4.33. La figure 4.34 montre les courbes débit-distorsion obtenues en évaluant la distorsion avec le PSNR et l'index SSIM. Les courbes obtenues avec le schéma AS2D sans post-traitement et JPEG2000 sont également illustrées. D'une manière générale, nous voyons que le schéma avec post-traitements (noté AS2D+Quadtree) améliore les résultats numériques de la technique AS2D. Cependant ils restent moins bons que ceux donnés par JPEG2000, à la fois au niveau du PSNR et de l'index SSIM. Pour *Lena* et *Cameraman*, ces résultats numériques sont aussi moins bons dans les bas débits que ceux obtenus à la section précédente avec une taille de maille $l_a = 16$.

Même si ces résultats numériques ne sont pas satisfaisants, il est important de les pondérer avec la qualité *visuelle* des images reconstruites. Nous pouvons faire les remarques suivantes. Jusqu'à $0,3$ bpp, la part de débit occupée par le maillage Quadtree est trop importante, malgré les améliorations, pour obtenir un gain par rapport à JPEG2000. Pour des débits allant de $0,3$ à $0,6$ bpp, la qualité de l'image reconstruite avec la méthode proposée peut apparaître meilleure qu'avec JPEG2000. Pour illustration, les figures 4.35 et 4.36 montrent les images reconstruites à $0,4$ bpp pour les quatre images tests. Avec la méthode AS2D, nous observons une reconstruction significativement plus nette des contours : les rebonds caractéristiques d'une compression par ondelettes ne sont quasiment plus présents. Certaines structures comme l'épaule, le chapeau de *Lena*, le manteau et les bâtiments dans *Cameraman*, le visage de *Barbara* et de manière générale les contours de *Peppers*, sont particulièrement bien rendues par

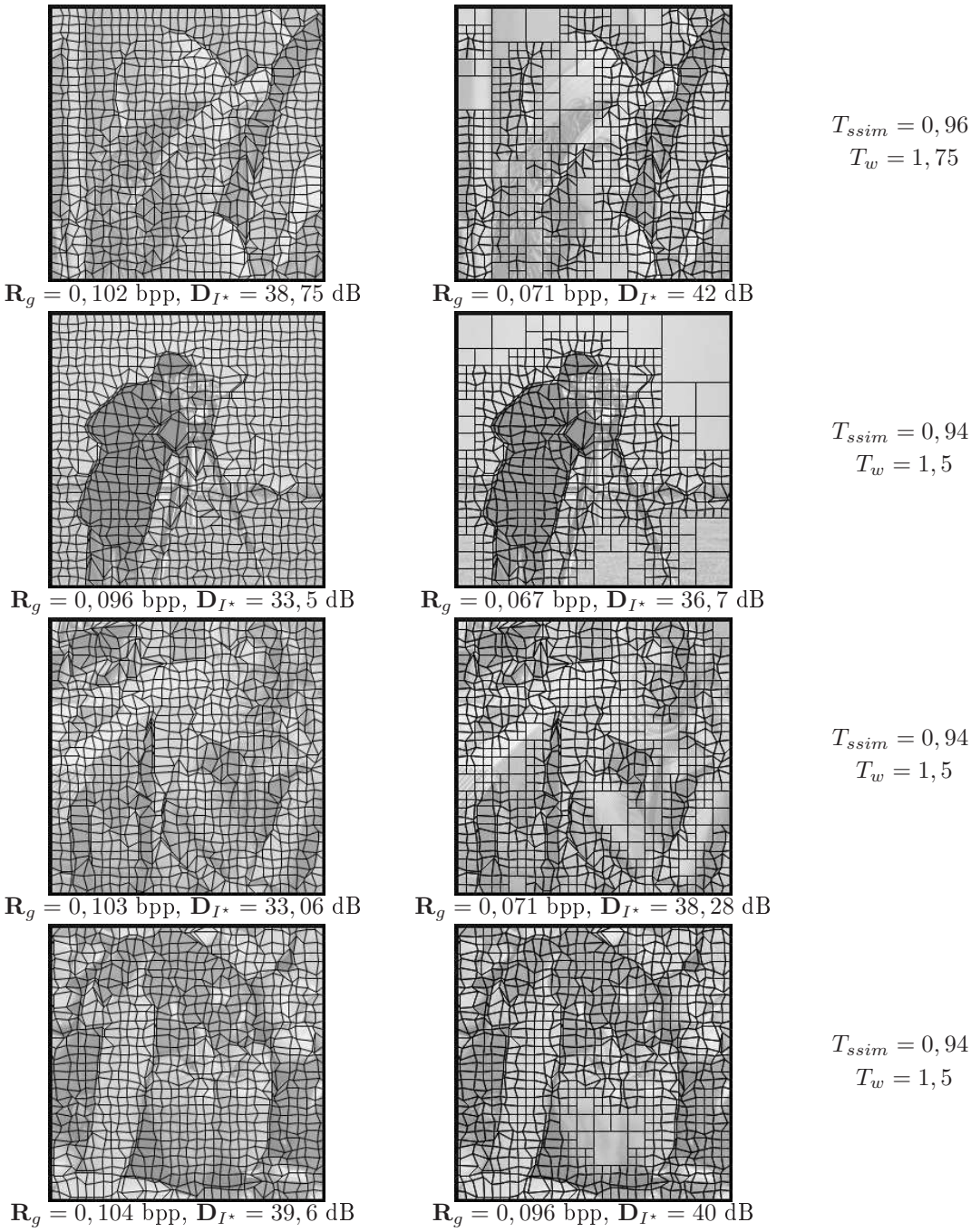


Fig. 4.33 : Création de maillages Quadtree à l'issue de l'analyse.

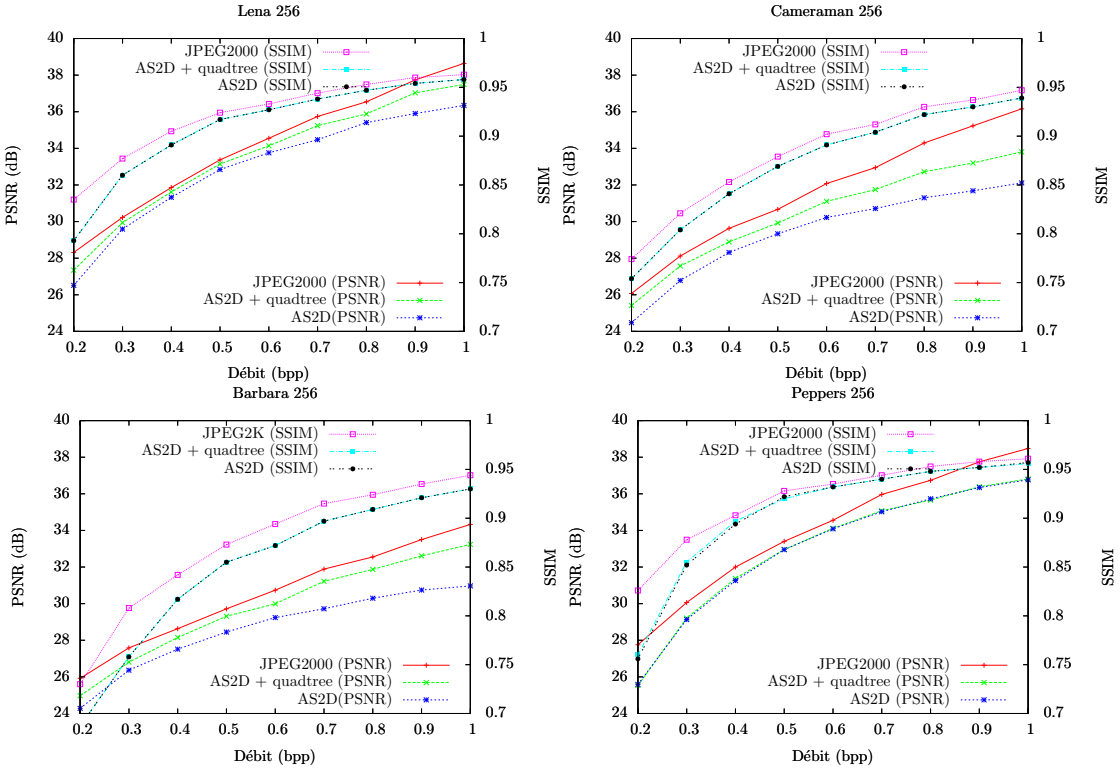


Fig. 4.34 : Résultats numériques de compression.

rapport à JPEG2000. Enfin, au-delà de $0,6$ bpp, le flou qui limitait la qualité visuelle des zones texturées par rapport à JPEG2000 dans la section précédente n'est plus perceptible, ceci grâce au post-traitement décrit au paragraphe 4.4.3.2.

Fig. 4.35 : Résultats de compression visuels à $0,4$ bpp.

Fig. 4.36 : Résultats de compression visuels à $0,4$ bpp.

4.5 Bilan du chapitre

Méthode proposée. Dans ce chapitre, nous avons décrit une nouvelle approche pour compresser une image fixe. La méthode proposée est de déformer le contenu spatial d'une image pour l'adapter à une décomposition en ondelettes séparables. La déformation étant modélisée par un maillage déformable partout connecté, une étape d'*analyse* permet de calculer la déformation qui minimise un *coût de description* de l'image déformée. Une technique d'optimisation de type descente en gradient a été proposée. L'estimation de la déformation spatiale est très semblable à une estimation de mouvement entre deux images d'une séquence vidéo. Après l'analyse, l'image déformée ainsi que les paramètres du maillage sont encodés et transmis. Après réception et décodage, une étape de *synthèse* permet de reconstruire l'image d'origine en inversant la déformation effectuée à l'analyse.

Distinction par rapport à l'art antérieur. La plupart des techniques antérieures tentent d'adapter le noyau d'ondelette à la géométrie d'une image. Dans notre technique, l'objectif est inversé car l'on cherche à adapter la géométrie (contours) d'une image au noyau. D'autres méthodes, comme les Bandelettes, avaient auparavant suivi une piste similaire. Cependant, ces méthodes sont basées blocs. Dans l'approche que nous avons proposée, la déformation est *globale* sur le domaine image et ceci permet d'éviter les effets de bords. D'autres travaux s'appuyant sur des maillages 2D avaient aussi été introduits précédemment. Cependant, dans ces travaux le maillage est utilisé comme grille d'échantillonnage pour l'image et non comme grille déformable, ce que nous proposons.

Résultats. Différents résultats ont été présentés dans ce chapitre. Nous avons tout d'abord étudié l'influence des paramètres d'analyse, du pas de quantification des nœuds du maillage et de la taille d'une maille.

Dans un premier temps, nous avons conclu qu'une taille de maille de l'ordre de 8×8 donnait un maillage trop cher à coder. Nous avons alors comparé les résultats de compression de notre schéma noté AS2D à ceux obtenus avec le standard JPEG2000 pour des images contenant une géométrie peu complexe. Dans ce cadre, nous avons noté des gains *visuels* dans les bas et moyens débits par rapport à JPEG2000. Dans les hauts-débits, les pertes dues aux ré-échantillonnages successifs lors de l'analyse puis de la synthèse produisent un flou dans les zones texturées et font chuter le PSNR. Pour améliorer la qualité visuelle de ces zones, nous avons alors proposé trois outils : *Quantification adaptative de l'image déformée*, *Encodage d'un résidu de synthèse*, *Augmentation de la résolution de l'image déformée*. Si la quantification adaptative n'a pas apporté de gain, les deux autres méthodes ont permis de relever les courbes dans les hauts-débits tout en conservant les performances à bas débits.

Dans un second temps, nous avons étudié la possibilité d'utiliser une taille de maille de l'ordre de 8×8 afin de modéliser des contenus géométriques plus complexes. Pour réduire le coût du maillage, nous avons proposé *trois post-traitements* à effectuer à l'issue de l'analyse. Le premier s'intéresse au cas des zones texturées. Puisque la déformation

des mailles dans ces zones a un coût non négligeable et génère de surcroît une perte visuelle dans les hauts débits, nous avons proposé d'« annuler » ces déformations en replaçant la maille sur le carré d'origine qu'elle occupait au début de l'analyse. Le second post-traitement propose de même de replacer sur leur carré d'origine toutes les mailles dont les déformations sont jugées non significatives. Ces deux post-traitements contribuent à améliorer la qualité des images reconstruites dans les hauts débits mais ne permettent pas de réduire suffisamment le coût du maillage. Nous avons alors proposé d'utiliser une structure en Quadtree pour regrouper des mailles carrées voisines et ainsi s'épargner le codage de nombreux symboles nuls. Cette technique a permis de réduire le coût du modèle de façon significative. Nous avons alors fourni un nouveau jeu de comparaisons avec JPEG2000 en considérant une taille de maille de l'ordre de 8×8. Si les résultats numériques sont moins bons que ceux fournis par le standard, les résultats visuels obtenus avec le schéma AS2D montrent des contours plus nets et une qualité visuelle générale qui nous semble meilleure dans les moyens débits. Dans les hauts débits, la qualité visuelle fournie par les deux schémas est très similaire.

Limites. Nous mettons en avant trois limites du schéma proposé. Tout d'abord, l'énergie minimisée ne prend pas en compte la distorsion de l'image reconstruite en bout de chaîne. Dans ces conditions, il est très difficile de mettre en place une optimisation débit-distorsion pour trouver les paramètres d'analyse et de quantification optimaux. En outre, puisque les métriques objectives comme le PSNR ou même l'index SSIM ne sont pas toujours en accord avec la qualité *visuelle* des images synthétisées, il est difficile de concevoir une énergie modélisant de façon juste la distorsion en bout de chaîne. Ensuite, le modèle de géométrie par maillage déformable partout connecté peut être remis en question. Nous avons en effet observé dans la première section que la contrainte de connectivité régulière imposée au maillage ne permet pas de satisfaire tous les objectifs énoncés en termes d'adaptation au contenu. En particulier, une maille ne peut tourner et s'étirer librement pour capturer la régularité le long des contours. Enfin, nous avons noté au cours de notre étude qu'une taille de maille de l'ordre de 8×8 était nécessaire pour pouvoir capturer suffisamment de caractéristiques géométriques. Or, dans ce cas le coût du maillage ne permet pas d'obtenir des résultats satisfaisants dans les bas débits.

Motivations pour une extension à la vidéo. Dans le chapitre qui suit, nous présentons un schéma par analyse-synthèse t+2D qui étend à la vidéo les principes introduits pour l'image fixe. Etant donné un groupe d'images (GOF) dans une séquence, nous proposons tout d'abord d'adapter son contenu temporel à un filtrage « en ligne » le long de l'axe temporel. Comme décrit dans des travaux précédents [TZ94a, WXCM99, Cam04b], ceci peut se faire en projetant toutes les images à un même instant de référence, après estimation et compensation en mouvement. Si cette projection permet un bon alignement temporel des images, alors les images projetées possèdent un contenu géométrique similaire. L'idée est de modéliser ce contenu géométrique une fois et une seule pour tout le GOF. En répartissant ainsi le coût de la géométrie sur plusieurs images, nous pouvons espérer que la part de débit occupée par cette information soit limitée et permette de reconstruire des contours de meilleure qualité visuelle tout en

conservant une bonne reconstruction des zones texturées. Cette idée est donc le point de départ des travaux que nous décrivons dans le chapitre suivant.

Chapitre 5

Adaptation spatio-temporelle d'un groupe d'images pour un codage par ondelettes t+2D

Les codeurs vidéo présentés au chapitre 3 prennent en compte les trajectoires de mouvement mais pas la géométrie des images (ni des sous-bandes temporelles dans le cas t+2D). Certains travaux [RAPP06] intègrent une dose de directionnalité au codage des images *intra* dans H.264/MPEG-4 AVC en modifiant le scan opéré dans les blocs pour l'obtention des coefficients DCT. Comme la géométrie évolue d'un instant à l'autre, il semble délicat dans ce type de schémas de ré-utiliser un même modèle de géométrie pour toutes les images d'un groupe d'images (GOF) : pour obtenir une modélisation précise de la géométrie des images inter, il est nécessaire d'estimer et de modéliser le flux géométrique pour chaque image. Le coût de l'information annexe (mouvement plus géométrie) s'avère alors trop lourd comparé au gain en qualité [RAPP07].

Dans ce chapitre, nous présentons une technique de codage par analyse-synthèse permettant de prendre en compte à la fois le mouvement et la géométrie dans un groupe d'images. Cette technique fusionne le codeur vidéo par analyse-synthèse temporelles proposé par Cammas [Cam04b] et la méthode d'adaptation spatiale décrite au chapitre précédent. L'idée principale du schéma présenté en section 5.1 est de déformer un GOF pour l'adapter à un filtrage séparable 3D le long des directions fixes horizontale, verticale et temporelle. Chaque image d'un GOF est alignée sur le même instant de projection après estimation et compensation en mouvement, puis **une même géométrie** est estimée pour chaque image compensée en mouvement. Dans la section 5.2, nous présentons les résultats de codage obtenus en modélisant à la fois la géométrie et les champs de mouvement par des maillages déformables partout connectés. Comme nous l'avons mentionné au chapitre 3 dédié au mouvement, ce modèle permet de reconstruire les images à la synthèse sans faire apparaître de pixels non connectés ou multiplement connectés, mais ne permet pas de modéliser les discontinuités de mouvement typiques des zones à occultation. Dans la dernière section, nous étudions les capacités de décorrélation temporelle de modèles moins contraints (modèle translationnel par blocs type

« Block Matching », modèle par blocs recouvrants $OBMC$, modèles hybrides $SCGI$ et $SOBMC$) du point de vue de l'*analyse* (bon alignement des images à l'instant de projection) comme du point de vue de la *synthèse* (bonne reconstruction des images en bout de chaîne).

5.1 Schéma proposé

5.1.1 Principe général

Le schéma de codage est illustré figure 5.1. Il généralise à la vidéo le schéma présenté au chapitre précédent. Le codeur proposé opère sur des GOF de taille N_G. Le but principal est de modifier le GOF, à travers une étape d'*analyse*, afin de réduire son coût de description dans une base séparable 3D non adaptative. A cause du mouvement et de la géométrie, les corrélations spatio-temporelles dans les images en entrée ne sont pas adaptées à une telle décomposition. L'analyse vise à compenser les images pour adapter les corrélations au filtrage horizontal, vertical, temporel : elle adapte le signal au noyau. Dans cette thèse, le noyau choisi est le noyau d'ondelette séparable 3D, mais l'approche peut être étendue à d'autres codeurs 3D (par exemple, la DCT-3D).

Après l'étape d'analyse, les images compensées sont appelées *textures*. Le terme texture est utilisé en référence au domaine de la modélisation 3D où un objet est synthétisé en plaquant une texture sur un maillage tridimensionnel. Le groupe de textures noté GOT peut être encodé par un simple codeur par ondelettes 3D sans recourir à aucune compensation : l'information de géométrie et de mouvement, que nous qualifierons dans la suite d'information de *structure*, est totalement découplée des textures à l'encodage. Les coefficients de texture et de structure sont donc quantifiés et codés *indépendamment*. Après décodage, le GOF d'origine peut être reconstruit en inversant la compensation spatio-temporelle opérée à l'encodage : c'est l'étape de *synthèse*.

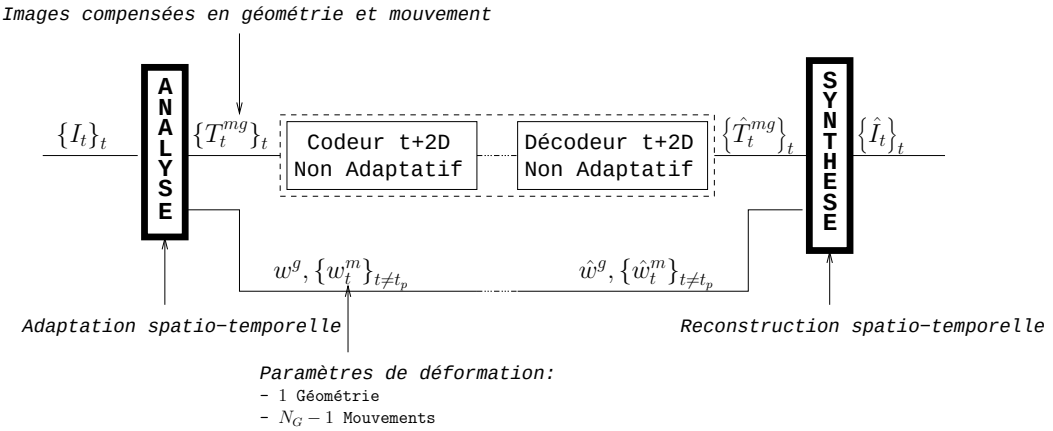


Fig. 5.1 : Méthode par Analyse-Synthèse t+2D. Le GOF en entrée est adapté aux directions de filtrage fixes horizontale, verticale et temporelle.

Les textures en sortie de l'analyse sont obtenues en effectuant une compensation en mouvement puis en géométrie des images en entrée. Pour cette section et la suivante, nous considérons que ces compensations sont définies par des transformations spatiales w *réversibles*. Le paragraphe suivant se concentre sur la brique d'analyse, pierre angulaire du schéma proposé.

5.1.2 Analyse

Considérons un GOF $\left\{I_{t_1}, \ldots, I_{t_{N_G}}\right\}$ de N_G images et définissons un *instant de projection* $t_p \in \{t_1, \ldots, t_{N_G}\}$. L'étape d'analyse prend ce GOF en entrée. En sortie, il génère trois informations :

- Un groupe de N_G textures $\{T_t\}_{t \in \{t_1, \ldots, t_{N_G}\}}$ noté GOT, images compensées en mouvement et en géométrie,
- Un groupe de $N_G - 1$ transformations $\{w_t^m\}_{t \neq t_p}$ donnant les correspondances de mouvement entre I_{t_p} et chaque image I_t pour $t \neq t_p$,
- Une transformation w_{BF}^g donnant les correspondances géométriques entre une texture cible (telle que définie au chapitre précédent) et une basse fréquence temporelle du GOT, notée I_{BF} et définie plus bas.

Ces informations sont générées en trois temps que nous décrivons maintenant.

5.1.2.1 « Alignement » temporel

L'étape d'alignement temporel correspond à l'analyse effectuée par Cammas [Cam04b]. Comme illustré figure 5.2, le but est d'aligner toutes les images sur l'image I_{t_p} de manière à obtenir un groupe d'images « sans » mouvement. Pour chaque instant $t \in \{t_1, \ldots, t_{N_G}\}, t \neq t_p$, une estimation de mouvement est effectuée entre I_{t_p} et I_t. Elle permet de calculer la fonction de mise en correspondances w_t^m entre t_p et t. Le critère utilisé pour l'estimation est l'erreur quadratique entre l'image I_{t_p} et l'image prédite notée $\bar{I}_{t_p \to t}(\mathbf{x}) = I_t(w_t^m(\mathbf{x}))$:

$$w_t^m = \arg\min_w \sum_{\mathbf{x} \in \mathcal{D}_{t_p}} [I_{t_p}(\mathbf{x}) - I_t(w(\mathbf{x}))]^2 \tag{5.1}$$

Cette erreur correspond à l'énergie $\mathbf{E}_i$ dans la méthode de Wang et Lee [WL94] présentée au chapitre 3. On parle aussi de *différence d'image déplacée* (DFD).

Dans les approches de lifting directionnel comme le Barbell lifting, les estimations de mouvement se font entre images voisines. Ici, toutes les images du GOF servent tour à tour de référence à l'image de projection. Au maximum, une distance de $N_G - 1$ images peut exister entre l'image à prédire à l'instant de projection et l'image de référence. Or, les algorithmes d'estimation vus au chapitre 3 recherchent les déplacements des pixels dans une fenêtre de taille limitée qui peut même être à l'échelle du pixel pour les techniques de descente en gradient. Lorsque l'image de référence est éloignée de l'image courante à prédire, il devient délicat d'estimer le mouvement avec précision sans donner une valeur initiale aux déplacements.

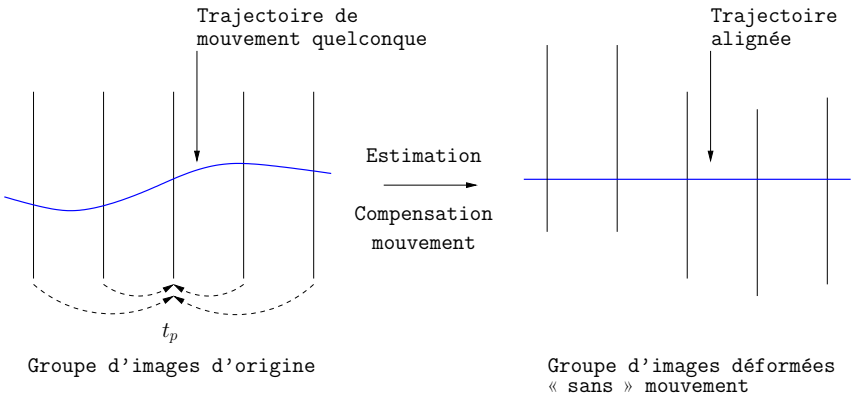


Fig. 5.2 : Illustration 1D de l'analyse temporelle. Les images du GOF sont projetées au même instant de projection t_p. A l'issue de cette projection, le groupe d'images déformées est décorrélé temporellement.

Pour résoudre ce problème, nous mettons en place un *suivi* de mouvement. Dans le cas où t_p est quelconque, un suivi de mouvement doit être effectué dans le sens causal entre t_p et $t = t_{N_G}$ et un suivi de mouvement doit être effectué dans le sens anti-causal entre t_p et $t = t_1$. Pour le suivi causal, le mouvement est tout d'abord estimé entre I_{t_p} et I_{t_p+1}, ce qui donne la transformation $w^m_{t_{p+1}}$. Étant donné un mouvement $w^m_{t_{p+k}}$ préalablement calculé, l'estimation du mouvement $w^m_{t_{p+k+1}}$ est alors initialisée avec les paramètres de $w^m_{t_{p+k}}$. Le suivi de mouvement anti-causal se déroule symétriquement.

Gestion des bords. Après alignement temporel (figure 5.3 à gauche), on remarque que les bords de certaines images n'ont pas de correspondants dans les autres images déformées le long de l'axe temporel. Pour éviter une gestion particulière des bords lors de la décomposition spatio-temporelle, les auteurs [WXCM99, Cam04b] proposent de définir les champs de mouvement w^m_t à l'instant t_p sur un domaine plus grand que le domaine image. Il faut s'assurer que l'extension du domaine soit suffisamment grande pour englober les déplacements de la caméra et des objets au cours du GOF. Dans les tests sur séquences CIF 30 Hz que nous présenterons, une extension de 16 pixels est effectuée aux bords de l'image I_{t_p} à l'instant de projection. Cette image étendue à l'instant de projection est souvent appelée *mosaïque*. Au début de l'analyse temporelle, les valeurs aux pixels des bords de la mosaïque ne sont pas définies. Après chaque estimation et compensation en mouvement, une nouvelle image $\bar{I}_{t_p \to t}$ alignée sur I_{t_p} est calculée et les valeurs aux bords de la mosaïque sont complétées avec les valeurs correspondantes dans $\bar{I}_{t_p \to t}$. Ceci permet d'améliorer les estimations de mouvement suivantes sur les bords. A la fin de l'analyse temporelle, les bords de toutes les images alignées sont complétés avec les valeurs correspondantes dans la mosaïque.

En général, la taille de l'extension choisie aux bords est toujours un peu plus grande que nécessaire. De ce fait, certaines zones (en noir sur la figure 5.3 à droite) restent à définir. Dans sa thèse [Cam04b] (chapitre 3), Cammas propose une technique d'ex-

trapolation, appelée « MR-pad » qui vise à limiter le coût de codage additionnel de l'extension. Pour une image donnée, cette technique permet de remplir les zones non définies par une prolongation régulière des zones définies. La méthode est itérative. Elle consiste à générer la pyramide multi-résolutions de l'image uniquement à l'aide des zones définies. Comme le signal obtenu est régulier aux bords, sa décomposition en ondelettes génère peu de hautes fréquences supplémentaires en comparaison des techniques d'extrapolation basiques (voir figure 5.4). Pour un groupe d'images, l'auteur utilise cette solution pour calculer les zones non définies sur les images extrêmes du GOF aligné ; les zones non définies des images intermédiaires sont ensuite calculées par interpolation des images extrêmes. Cette interpolation permet de limiter l'énergie des hautes fréquences temporelles aux bords. Notons qu'une extrapolation spatio-temporelle plus complexe par analyse-synthèse 3D donnant de meilleures performances est aussi proposé par l'auteur mais nous ne l'avons pas implémentée ici.

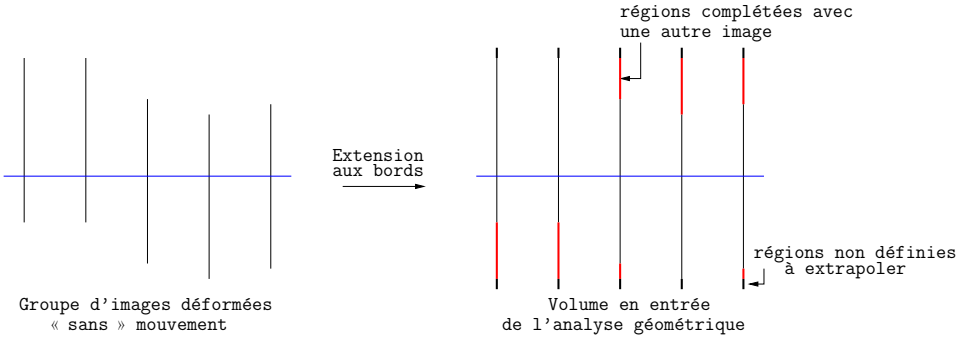


Fig. 5.3 : Pour effectuer un filtrage « en ligne » sans gestion particulière des bords, il faut recourir à une extrapolation des images compensées.

5.1.2.2 Décorrélation géométrique

Le but de la deuxième étape de l'analyse est d'estimer **une même géométrie** pour toutes les images compensées en mouvement $\bar{I}_{t_p \to t}$, $t \in \{1 \dots N_G\}$. Dans l'hypothèse que l'analyse du mouvement a permis un alignement précis sur l'instant de projection, toutes les images du GOF compensé ont une géométrie proche. Ici, nous définissons le signal I_{BF} comme l'image moyenne du GOF compensé, à savoir :

$$I_{BF}(\mathbf{x}) = \frac{1}{N_G} \sum_{t=1}^{N_G} \bar{I}_{t_p \to t}(\mathbf{x}), \quad \mathbf{x} \in \mathcal{D}_{t_p} \tag{5.2}$$

Si l'alignement temporel est suffisamment précis, la géométrie de I_{BF} est très proche de la géométrie de chaque image compensée en mouvement. Nous proposons donc de calculer la géométrie uniquement sur I_{BF}. Pour ce faire, nous utilisons la technique détaillée au chapitre précédent. Le but est de trouver la transformation notée w^g_{BF} qui permette de minimiser le coût de description de la texture T_{BF} définie par

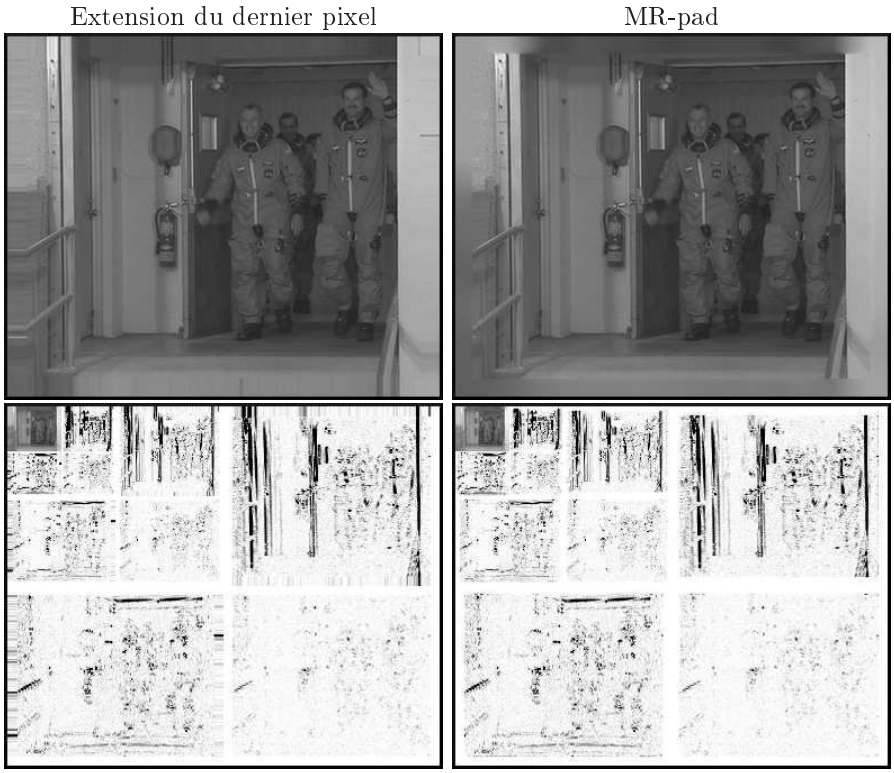


Fig. 5.4 : Extension de 16 pixels aux bords de la première image de la séquence *Crew*. Avec le « MR-pad » proposé dans [Cam04b], l'énergie des hautes fréquences d'ondelettes **sur les bords** est limitée comparée à une extension par prolongement du dernier pixel.

$T_{BF}(\mathbf{u}) = I_{BF}(w^g_{BF}(\mathbf{u})) \quad \forall \mathbf{u} \in \tilde{\mathcal{D}}_{t_p}$. Nous avons montré que cette déformation est celle qui minimise la DFD entre l'image I_{BF} et une texture cible inconnue :

$$w^g_{BF} = \arg\min_{w} \sum_{\mathbf{u}\in\tilde{\mathcal{D}}_{t_p}} [I_{BF}(w(\mathbf{u})) - T_{cible}(\mathbf{u})]^2 \tag{5.3}$$

En observant les équations (5.1) et (5.3), nous remarquons que les fonctions de coût que l'on cherche à minimiser lors de l'estimation de mouvement et de l'estimation de géométrie sont très proches. Nous rappelons que ceci nous a permis dans le chapitre précédent (page 124) de présenter une technique d'estimation géométrique ressemblant très fortement à une estimation de mouvement.

5.1.2.3 Génération du GOT

Le groupe des textures compensées **en mouvement *et* en géométrie** $\{T_t\}$, $t \in \{1, \ldots, N_G\}$, est finalement obtenu en combinant la compensation spatiale w^g_{BF} avec chaque compensation en mouvement w^m_t, $t \neq t_p$. Cette *adaptation spatio-temporelle* des images est appliquée aux images d'origine du GOF. Si I_t est une de ces images, alors la texture résultante se calcule suivant l'opération :

$$T_t(\mathbf{u}) = I_t(w^g_{BF} \circ w^m_t(\mathbf{u})) \quad \forall \mathbf{u} \in \tilde{\mathcal{D}}_{t_p} \tag{5.4}$$

Dans la suite nous noterons w^{mg}_t la combinaison de la compensation géométrique w^g_{BF} avec la compensation en mouvement w^m_t :

$$w^{mg}_t(\mathbf{u}) = w^g_{BF} \circ w^m_t(\mathbf{u}) \quad \forall \mathbf{u} \in \tilde{\mathcal{D}}_{t_p} \tag{5.5}$$

5.1.3 Encodage

A l'issue de l'analyse, trois types d'information sont à coder et transmettre : le GOT, la géométrie w^g_{BF} et les $N_G - 1$ mouvements $\{w^m_t\}_{t\neq t_p}$. L'encodage de chaque type d'information est réalisé *indépendamment.*

Les résultats donnés dans la suite ont été obtenus en effectuant un encodage t+2D du GOT. Le GOT est tout d'abord décomposé temporellement à l'aide de l'ondelette 5/3. Ce choix est motivé par les résultats donnés dans [Cam04b]. Ensuite, les sous-bandes temporelles générées sont envoyées au codeur JPEG2000 pour décomposition spatiale et encodage. L'ondelette choisie pour la décorrélation spatiale est l'ondelette de Daubechies 9/7 [ABMD92]. La quantification opérée par JPEG2000 est le fruit d'un compromis débit-distorsion sur l'ensemble des sous-bandes spatio-temporelles. Comme pour l'image fixe, l'option d'encodage « scalable » est activée.

Pour la géométrie, le modèle par maillage déformable est utilisé. L'encodage de w^g_{BF} suit le procédé choisi au chapitre précédent : les déplacements des nœuds par rapport au maillage uniforme dans $\tilde{\mathcal{D}}_{t_p}$ sont quantifiés et codés en plans de bits avec un codeur arithmétique.

Pour le mouvement, le modèle par maillage déformable est utilisé dans la section suivante, puis d'autres modèles (BM, OBMC, SOBMC, SCGI) sont testés en section 5.3.

Pour le maillage déformable, deux modes de codage seront comparés. Les paramètres des déformations w_t^m pour chaque instant $t \neq t_p$ sont les positions $\mathbf{x}_i^t$ des sommets du maillage. Le premier mode de codage consiste à prédire la position d'un nœud $\mathbf{x}_i^t$ avec la position du nœud à l'instant précédent si $t > t_p$ ou à l'instant suivant si $t < t_p$, et de quantifier et coder le résidu de prédiction. Le second mode de codage consiste à considérer les positions d'un nœud i à chaque instant $\left\{\mathbf{x}_i^t \quad | \quad t \neq t_p\right\}$ comme un signal 1D, à effectuer une décomposition en ondelette sur ce signal 1D, puis à quantifier et coder les coefficients obtenus. La comparaison de ces deux modes de prédiction est faite au paragraphe 5.2.2. Pour la dernière section, le premier mode de codage a été retenu. Lorsque le modèle de mouvement (SOBMC ou SCGI) nécessite un label (valeur binaire) de connectivité en chaque nœud, ce label est encodé à l'aide d'un codeur arithmétique, sans contexte particulier. Les modèles de mouvement et les techniques d'estimation associées ont été décrites au chapitre 3.

5.1.4 Synthèse

Après transmission des informations, le récepteur décode les textures, la géométrie, et les mouvements en inversant les étapes d'encodage décrites précédemment. En notant $\left\{\hat{T}_t\right\}_{t\in\left\{t_1\ldots t_{N_G}\right\}}$ les textures décodées, $\hat{w}_{BF}^g$ la géométrie décodée et $\{\hat{w}_t^m\}_{t\in\{t_1\ldots tN_G\},t\neq t_p}$ les mouvements décodés, la synthèse de chaque image d'origine est réalisée en inversant la compensation spatio-temporelle réalisée à l'analyse :

$$\begin{aligned}\hat{I}_t(\mathbf{x}) &= \hat{T}_t[(\hat{w}_{BF}^g \circ \hat{w}_t^m)^{-1}(\mathbf{x})] \\ &= \hat{T}_t[(\hat{w}_t^{mg})^{-1}(\mathbf{x})], \quad \forall \mathbf{x} \in \mathcal{D}_t\end{aligned} \tag{5.6}$$

Dans la section suivante, nous allons reprendre chaque étape du schéma en utilisant le maillage déformable à la fois pour modéliser la géométrie et les champs de mouvement. Des résultats de codage seront présentés dans le dernier paragraphe et comparés à des travaux antérieurs.

5.2 Résultats avec une modélisation de la géométrie et du mouvement par maillage déformable

5.2.1 Analyse-Synthèse : illustrations

5.2.1.1 Alignement temporel

Le maillage choisi pour modéliser le mouvement est un maillage quadrangulaire. Lors de l'étape d'estimation de mouvement, un maillage uniforme est placé à l'instant de projection t_p et les positions des nœuds à chaque instant $t \neq t_p$ qui minimisent (5.1) sont recherchées. Pour modéliser le mouvement, nous choisirons toujours dans la suite une taille d'arête $l_a = 16$. La technique de descente en gradient détaillée en annexe B est utilisée pour chaque nouvelle estimation. Rappelons que cette technique est itérative.

Nous notons k_{max} le nombre d'itérations utilisé. Après chaque itération de la descente en gradient, la technique de projection non-obtuse est appliquée pour s'assurer que toutes les mailles restent conformes.

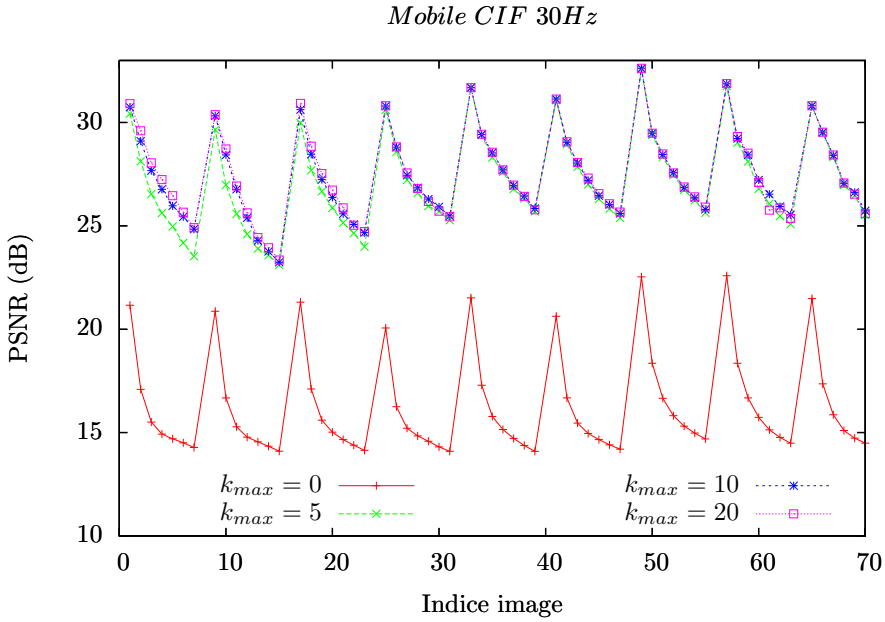


Fig. 5.5 : Efficacité de l'alignement temporel après estimation et compensation en mouvement. Le PSNR affiché est le PSNR entre l'image I_{t_p} à l'instant de projection et chaque image compensée en mouvement $\bar{I}_{t_p \to t}$, $t \neq t_p$.

Efficacité de l'alignement temporel. Lors de chaque estimation de mouvement entre la mosaïque I_{t_p} et une image I_t du GOF ($t \neq t_p$), le critère minimisé (5.1) est l'erreur quadratique entre I_{t_p} et l'image compensée $\bar{I}_{t_p \to t}$. Le PSNR entre ces deux images est donc une mesure juste pour évaluer l'efficacité de l'alignement temporel. Sur la figure 5.5, nous affichons les PSNR obtenus en appliquant l'analyse temporelle à la séquence *Mobile CIF* $30Hz$ et en faisant évoluer le nombre d'itérations k_{max}. La taille des GOF N_G choisie est 8 et l'instant de projection est le premier instant de chaque GOF. Puisque les images aux instants de projection ne sont pas modifiées lors de la compensation en mouvement, leur PSNR (infini) n'est pas représenté sur la figure. Sur cette figure, nous remarquons qu'après seulement 5 itérations de la descente en gradient, les PSNR augmentent de près de 10 dB. Ceci démontre l'efficacité de l'optimisation et du suivi. Logiquement, plus on s'éloigne de l'instant de projection, moins l'alignement est bon. Ceci s'explique principalement par le fait que certaines zones présentes à l'instant de projection disparaissent au fur et à mesure que l'on s'éloigne de cet instant. Ces zones sont donc de moins en moins bien prédites. Avec 10 itérations, des gains sont observés sur certains GOF. Au-delà, les résultats sont sensiblement identiques. Dans la suite, nous nous limiterons donc à 10 itérations pour chaque estimation de mouvement.

La figure 5.6 montre les mouvements des nœuds du maillage estimés avec $k_{max} = 10$.

Le but de l'alignement temporel est d'adapter le contenu du GOF à une décomposition « en ligne » le long de l'axe temporel. Sur la figure 5.7, nous avons représenté la basse fréquence temporelle et la première couche de détails obtenues après décomposition en ondelettes du GOF compensé le long de l'axe temporel. L'ondelette utilisée est l'ondelette 5/3. Le résultat est comparé avec celui obtenu sans compensation en mouvement ($k_{max} = 0$). Comme on peut le voir, la basse fréquence obtenue sans compensation en mouvement présente un effet fantôme généralisé sur tout le domaine image. Avec compensation en mouvement, ce phénomène n'est plus visible à l'œil nu, ce qui démontre le bon alignement des images. En conséquence, l'énergie dans la couche de rehaussement est sensiblement moins importante : nous avons adapté le contenu des images à une décomposition selon l'axe temporel.

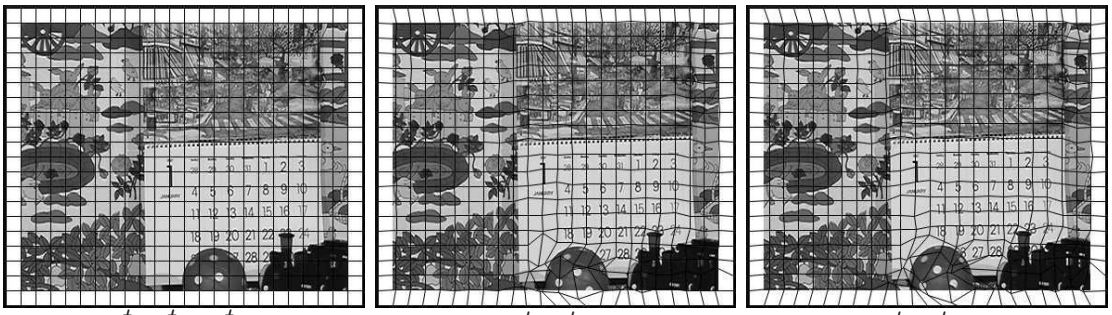


Fig. 5.6 : Suivi de mouvement obtenu après $k_{max} = 10$ itérations de descente en gradient pour chaque image.

Choix de l'instant de projection t_p. Comme nous venons de le voir, la qualité de la compensation en mouvement pour une image donnée dépend de la distance entre cette image et la mosaïque. En plaçant l'instant de projection en milieu de GOF, on améliore le PSNR moyen des images compensées par rapport à I_{t_p}. Ce résultat est illustré sur la figure 5.8. Nous verrons que placer l'instant de projection en milieu de GOF permet également d'améliorer la qualité des images reconstruites en bout de chaîne.

5.2.1.2 Modélisation géométrique

Si l'alignement temporel a été efficace, toutes les images compensées en mouvement se ressemblent fortement et présentent une géométrie similaire. Plutôt que de modéliser et d'estimer une géométrie pour chacune d'entre elles, nous estimons donc une seule géométrie pour tout le GOF. Cette géométrie est estimée sur une image basse fréquence I_{BF} qui est simplement la moyenne du GOF compensé en mouvement (définition (5.2)). Notons que rien n'oblige à utiliser une même taille de maille pour modéliser la géométrie et le mouvement. Dans les travaux présentés dans ce chapitre, nous avons utilisé une taille $l_a = 8$ pour modéliser la géométrie (donc deux fois plus petite celle utilisée pour modéliser le mouvement). La figure 5.9 montre en haut les images I_{BF} obtenues sur

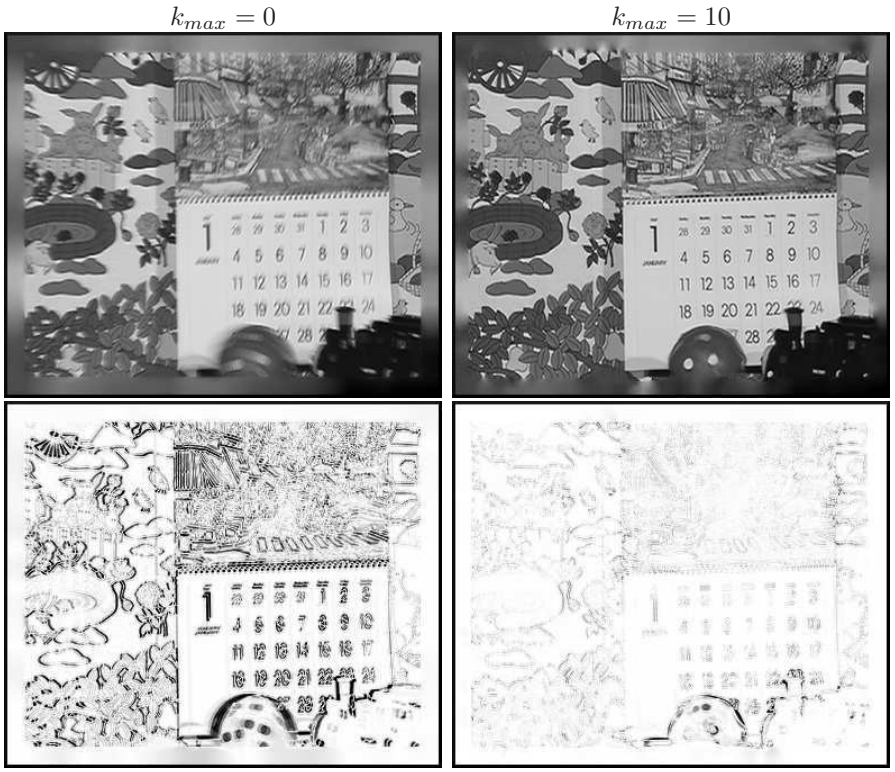


Fig. 5.7 : A gauche, basse fréquence temporelle et première couche de rehaussement obtenue après décomposition du GOF d'origine le long de l'axe temporel. A droite, basse fréquence temporelle et première couche de rehaussement obtenue après décomposition du GOF compensé le long de l'axe temporel.

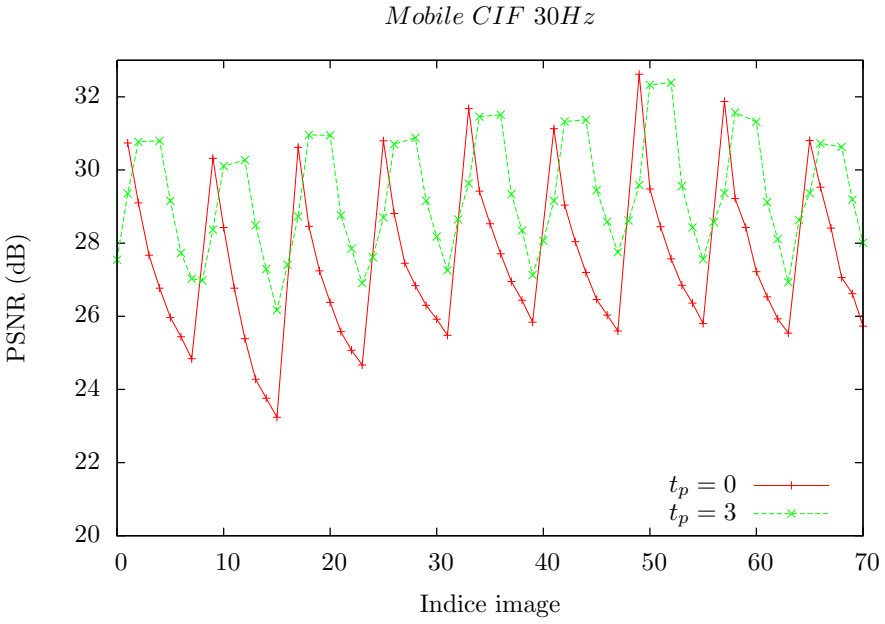


Fig. 5.8 : Placer l'instant de projection en milieu de GOF permet d'améliorer l'alignement des images compensées en mouvement.

les trois premiers GOF de la séquence *Foreman CIF* $30Hz$ et en bas le résultat de l'estimation de géométrie, telle que nous l'avons présentée au chapitre précédent. Les post-traitements pour améliorer la reconstruction des textures et limiter les déformations non significatives sont utilisés en prenant les seuils $T_{ssim} = 0,96$ et $T_w = 1,5$.

Sur la figure 5.9, nous remarquons que les images de basse fréquence présentent peu d'effet fantôme, preuve d'un bon alignement temporel. Certaines zones à occultation sont cependant moins bien prédites que d'autres, comme l'œil droit de *foreman* dans le premier GOF, ou le haut de son chapeau dans le deuxième GOF. Si l'on observe le résultat de l'estimation de géométrie, on remarque que de différents contours ont été capturés, par exemple ceux du visage ou des arêtes du bâtiment dans le fond.

5.2.1.3 Génération du GOT

A l'issue des étapes précédentes, nous disposons :

- d'un groupe d'images compensées en mouvement dont le contenu est adapté à une décomposition 1D le long de l'axe *temporel*,
- d'un modèle de déformation géométrique dont le but est d'adapter le contenu *spatial* de toutes les images du GOF compensé à une décomposition 2D séparable (horizontale-verticale) par ondelettes.

Pour générer le groupe des textures adapté à un filtrage 3D horizontal-vertical-temporel fixe, la déformation géométrique peut être appliquée à toutes les images du GOF compensé en mouvement. Cependant, nous préférons générer les textures directe-

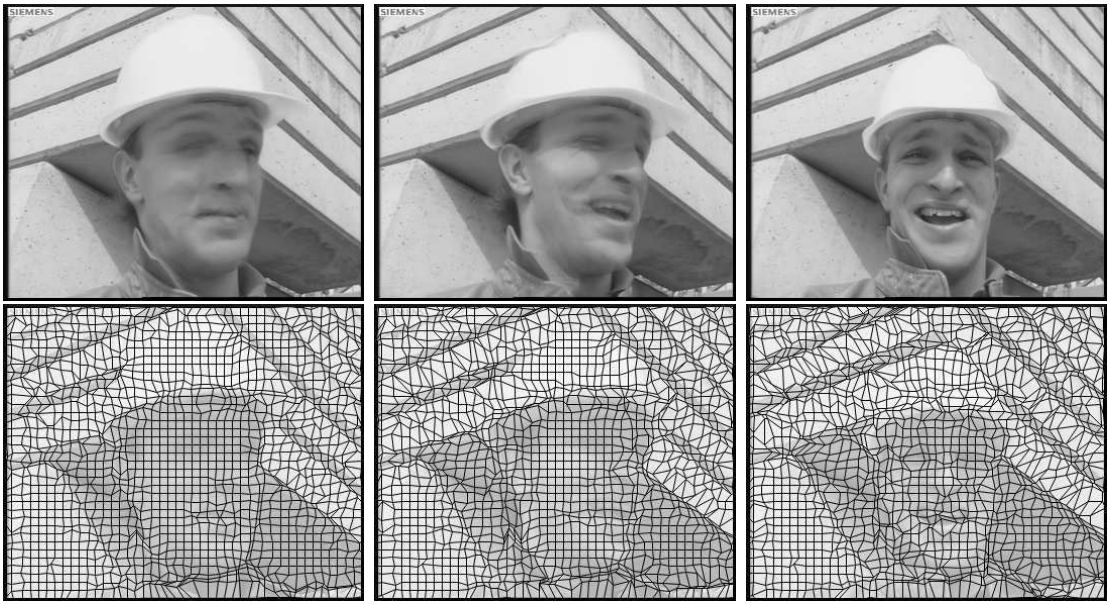


Fig. 5.9 : En haut, images basse fréquence I_{BF} obtenues après alignement temporel pour les trois premiers GOF de la séquence $Foreman\ CIF\ 30Hz$. En bas, modèle de géométrie estimé sur chacune de ces images. **Ce modèle est celui utilisé pour toutes les images du GOF compensé en mouvement.**

ment à partir des images d'origine en utilisant l'équation (5.4). Ceci permet de ramener à 1 le nombre de ré-échantillonnage des images d'origine et ainsi limiter les pertes. Sur la figure 5.10, nous illustrons la création d'une texture à partir d'une image d'origine au dernier instant t_7 du premier GOF de la séquence $Foreman\ CIF\ 30Hz$. Un pixel $\mathbf{u}$ du domaine texture $\tilde{\mathcal{D}}_{t_7}$ est mis en correspondance avec une position $\mathbf{x}'$ dans le domaine de l'image compensée en mouvement $\bar{I}_{t_p \to t_7}$. Cette correspondance est donnée par la déformation géométrique : $\mathbf{x}' = w^g_{BF}(\mathbf{u})$. La position $\mathbf{x}'$ est ensuite mise en correspondance avec une position dans le domaine image $\mathcal{D}_{t_7}$ d'origine. Cette correspondance est donnée par la déformation temporelle $w^m_{t_7}$: $\mathbf{x} = w^m_{t_7}(\mathbf{x}')$. La valeur interpolée de l'image au point $\mathbf{x}$ donne la valeur du pixel $\mathbf{u}$ de la texture.

En observant l'image compensée en mouvement sur la figure 5.10, on remarque que certaines régions comme le nez ou le bord droit du visage et du chapeau ont été étirées. Ces étirements de texture interviennent lorsqu'une zone *disparaît* entre l'instant de projection et l'instant courant (ici t_7). A l'opposé certaines régions, principalement le bord gauche du visage, se trouvent contractées lors de la compensation en mouvement. Ces contractions de texture interviennent lorsqu'une zone *apparaît* entre l'instant de projection et l'instant courant. Comme nous l'avons décrit au chapitre précédent, un étirement de texture s'accompagne d'un gain de résolution par rapport à l'instant courant, tandis qu'une contraction s'accompagne d'une perte de résolution. L'image de texture obtenue montre quant à elle des étirements de texture au niveau de nombreux contours : l'adaptation spatiale a tendance à lisser les discontinuités pour réduire l'éner-

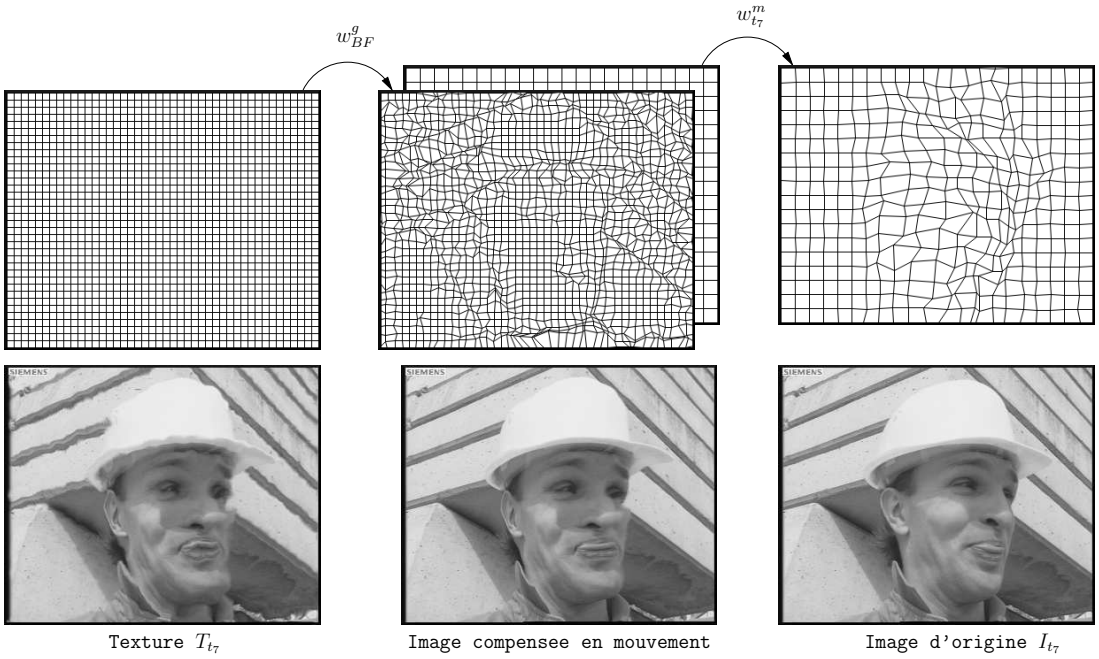


Fig. 5.10 : Illustration du procédé de création d'une texture à partir d'une image d'origine I_{t_7}, son mouvement $w^m_{t_7}$ par rapport à t_p et sa géométrie w^g_{BF}.

gie des coefficients d'ondelettes après décomposition horizontale-verticale. Comme on le voit, les textures obtenues à l'issue de l'analyse peuvent avoir un contenu singulier. Il est important de souligner que ces textures ne sont pas destinées à être visualisées. Elles sont encodées et transmises avec la déformation géométrique et les déformations temporelles pour pouvoir synthétiser une version des images d'origine en bout de chaîne.

5.2.1.4 Synthèse

En modélisant géométrie et mouvement par un maillage déformable, toutes les déformations effectuées à l'analyse sont inversibles. Chaque image du GOF d'origine peut donc être synthétisée en suivant l'équation (5.6). Comme dans le cas de l'image fixe, l'aller retour entre le domaine image et le domaine texture introduit des pertes numériques irréversibles. En compensant les images en mouvement, nous introduisons de nouvelles pertes de résolution par rapport au chapitre précédent : lorsqu'une zone apparaît entre l'instant de projection et l'instant courant, elle se retrouve contractée lors de la compensation comme expliqué précédemment. Pour limiter ces pertes de résolution, nous pourrions transmettre des textures plus grandes que les images d'origine ou encore transmettre des images de résidus, de la même façon que nous l'avons testé pour l'image fixe. Dans les résultats que nous présentons dans ce chapitre, nous n'avons cependant pas intégré de traitement spécifique pour ces zones. Le seul paramètre avec lequel nous avons tenté de limiter les pertes de résolution est l'instant de projection

t_p. La figure 5.11 supporte cette affirmation. Elle montre le PSNR et l'index SSIM des images reconstruites en bout de chaîne en plaçant l'instant de projection en début de GOF $t_p = t_0$ et en milieu de GOF $t_p = t_3$. Comme on le voit, positionner t_p en milieu de GOF permet de limiter les sauts de qualité numérique à l'intérieur d'un GOF. En effet, en plaçant t_p en milieu de GOF, on limite la distance moyenne entre la mosaïque et les autres images du GOF et donc l'apparition de nouvelles zones.

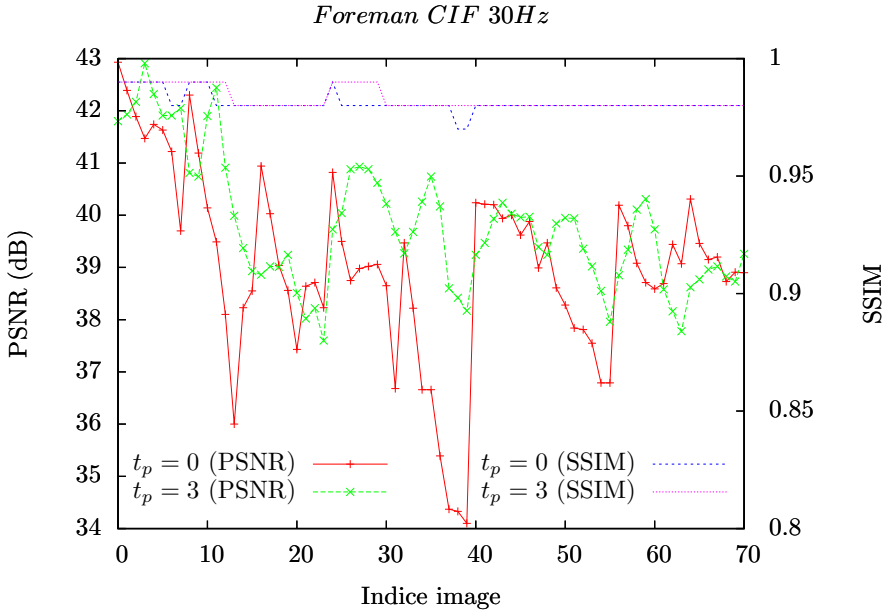


Fig. 5.11 : Placer l'instant de projection en milieu de GOF permet d'améliorer la qualité moyenne des images synthétisées.

Sur la figure 5.11, on remarque que le PSNR des images synthétisées avec $t_p = t_3$ évolue entre 38 et 43 dB, tandis que l'index SSIM évolue entre $0,98$ et $0,99$. Avec de telles valeurs, les différences avec les images d'origine sont difficilement perceptibles.

5.2.2 Encodage

Encodage de la géométrie. L'encodage de la géométrie w^g_{BF} suit le même procédé que celui utilisé au chapitre précédent. Les déplacements des nœuds du maillage sont quantifiés puis codés en utilisant un codage en plans de bits. Les résultats donnés dans ce chapitre ont été obtenus en utilisant un pas de quantification égal à 1 (précision pixellique).

Encodage du mouvement. Nous nous penchons ici sur le codage des champs de mouvement $\{w^m_t\}_{t \neq t_p}$. Dans les codeurs standards, le mouvement d'un bloc est estimé avec une précision maximale donnée (pixel, 1/2, ou 1/4 pixel). Il peut ensuite être prédit par le mouvement d'un bloc voisin avant encodage, mais il n'y a pas de transformée

temporelle. Dans [Cam04b], le mouvement est modélisé par un maillage et les positions sont estimées avec une précision flottante. Comme le maillage modélise un champ de mouvement continu et que le mouvement est suivi sur tout un GOF, l'auteur propose de décomposer temporellement les positions de chaque nœud du maillage à l'aide d'une ondelette 5/3, puis de quantifier les coefficients transformés. La question que nous nous posons est la suivante : pour une même distorsion sur le mouvement, réduit-on son coût en décomposant les déplacements avec une ondelette avant quantification et codage ?

Nous testons ici les deux méthodes sur les ensembles de déplacements issus de l'analyse temporelle des trois premiers GOF de *foreman*. Dans la première méthode, la position $\mathbf{x}_i(t)$ d'un nœud à un instant $t \neq t_p$ est prédite par la position à l'instant $t-1$ si $t > t_p$ ou $t+1$ si $t < t_p$. Les résidus de prédictions sont quantifiés puis encodés en plans de bits à l'aide d'un codeur arithmétique. Chaque ensemble de positions $\{\mathbf{x}_i(t)\}$ pour un instant t est codé séparément et les statistiques du codeur sont ré-initialisées pour chaque groupe. Dans la deuxième méthode, des résidus de prédiction sont calculés comme précédemment. Les différents résidus de prédictions $\Delta\mathbf{x}_i(t)$ d'un nœud i sont ensuite décomposés temporellement à l'aide d'une ondelette. Les ondelettes 9/7 et 5/3 sont testées. Les coefficients d'ondelettes sont quantifiés puis encodés en plans de bits à l'aide d'un codeur arithmétique. Les statistiques du codeur sont ré-initialisées pour chaque sous-bande temporelle. En faisant évoluer le pas de quantification de $0,25$ à 32 pour les deux méthodes, nous obtenons les courbes de la figure 5.12. Comme on peut l'observer, quantifier les déplacements dans le domaine spatial donne les meilleurs performances. Ceci rejoint nos conclusions du chapitre précédent sur le codage de la géométrie.

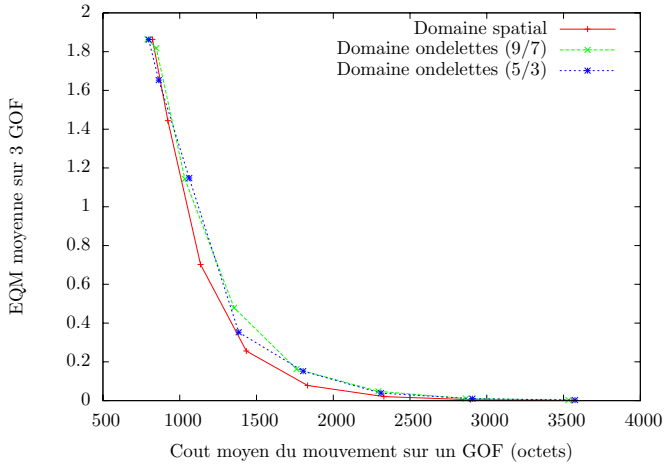


Fig. 5.12 : Quantifier les déplacements dans le domaine spatial aboutit à un meilleur compromis débit-distorsion que les quantifier dans le domaine ondelettes.

Encodage des textures. Comme pour le schéma AS2D du chapitre précédent, les textures sont générées *avec les maillages quantifiés de plus haute qualité*. Ces textures

sont adaptées à une décomposition ondelettes 3D selon les axes temporel, horizontal et vertical. Nous proposons de les décomposer selon un schéma t+2D avant encodage. Dans un premier temps, nous appliquons la décomposition temporelle. Dans [Cam04b], l'auteur montre que les ondelettes 9/7 et 5/3 aboutissent à des performances similaires en termes débit-distorsion (et meilleures que l'ondelette 5/3 tronquée). Nous avons utilisé l'ondelette 5/3 dans nos résultats. Cette décomposition temporelle génère une image de basse fréquence temporelle et $N_G - 1$ images de haute fréquence. Ces N_G images sont ensuite envoyées au logiciel JPEG2000 (VM 8.0) qui se charge d'effectuer la décomposition horizontale verticale de chaque sous-bande puis de générer un flux binaire via *EBCOT*. Dans les tests présentés plus bas, la base d'ondelettes choisie pour décomposition spatiale est l'ondelette 9/7. Comme dans les travaux du chapitre précédent, nous activons l'option `-Clayers` de JPEG2000 qui permet de générer un flux scalable. JPEG2000 optimise le compromis débit-distorsion sur l'ensemble des sous-bandes temporelles.

A l'issue de l'encodage, trois flux scalables ont été créés, pour la géométrie, le mouvement et les textures. Nous allons maintenant présenter les résultats de compression du schéma AS2D+t. Ces résultats sont comparés avec le schéma AS t obtenu en désactivant la compensation géométrique ainsi qu'avec le codeur H.264/MPEG-4 SVC.

5.2.3 Résultats de compression

Cette section présente les résultats de compression obtenus en utilisant le maillage déformable comme modèle de géométrie et de mouvement. Avant de comparer le schéma AS2D+t au standard, nous proposons d'étudier l'influence du pas de quantification utilisé pour le mouvement sur les performances débit-distorsion. Nous donnons ensuite des résultats de compression comparatifs. Une des questions principales est de savoir si la prise en compte de la géométrie apporte un gain par rapport à un schéma du type AS t où seul le mouvement est pris en compte pour générer les textures. Nous comparons également nos résultats numériques et visuels au standard H.264/MPEG-4 dans sa version scalable SVC (JSVM 8.9). Enfin, comme dans le chapitre précédent, nous évaluons l'impact d'un décodage avec perte de la structure (géométrie et/ou mouvement) sur les résultats de compression.

5.2.3.1 Précision du mouvement

Nous étudions ici l'influence du pas de quantification utilisé pour quantifier les champs de mouvement avant création et encodage du GOT. Nous notons Q_m ce pas et $\mathbf{R}_m$ la bande passante en kb/s occupée par l'information de mouvement. Nous travaillons sur les 10 premiers GOF (80 premières images) de la séquence *Foreman CIF* $30Hz$. Pour ces tests, la compensation géométrique est désactivée. Pour chaque pas $Q_m \in \{0.25, 0.5, 1, 2\}$, nous encodons le mouvement et les images compensées en mouvement. Après réception, le mouvement est décodé sans perte et les images compensées sont décodées de façon à atteindre les débits cibles 128, 256, 512, 1024 kb/s. Nous calculons

le PSNR moyen des images reconstruites. Les résultats sont présentés sur la figure 5.13. Le débit moyen $\mathbf{R}_m$ occupé par le mouvement est indiqué pour chaque pas de quantification. Comme on peut l'observer, plus le pas de quantification est grand, moins bonnes sont les performances dans les hauts débits et meilleures sont les performances dans les très bas débits. Comme dans le cas de la géométrie, un pas de quantification égal à 1 semble un bon compromis pour parcourir une large gamme de débits. Nous avons donc fixé Q_m à 1 pour générer les résultats présentés au paragraphe suivant.

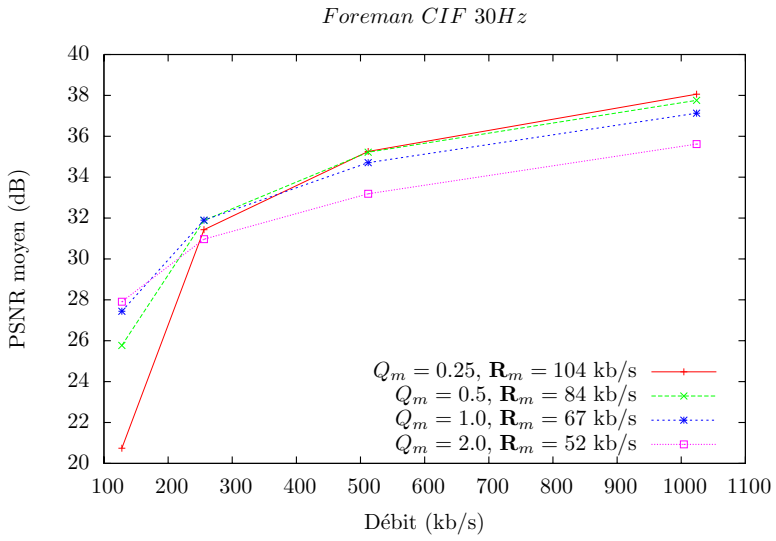


Fig. 5.13 : Influence de la précision du mouvement sur les performances débit-distorsion.

5.2.3.2 Comparaison AS2D+t, AS t, H.264/MPEG-4 SVC

Dans ce paragraphe nous présentons les résultats de compression obtenus avec notre schéma d'analyse-synthèse spatio-temporelles, noté AS2D+t. Nous les comparons avec le schéma d'analyse-synthèse temporelles, noté AS t, où le GOT encodé et transmis est le groupe d'images compensées en mouvement. Pour le schéma AS t, aucune géométrie n'est prise en compte ni transmise et on se ramène donc à un codeur similaire à celui proposé dans [Cam04b]. Puisque tous les flux encodés dans les schémas AS2D+t et AS t sont « scalables », le standard de compression auquel nous nous comparons est le standard H.264/MPEG-4 dans sa version scalable SVC. Les vidéos utilisées dans nos tests sont *Foreman*, *Akiyo*, *Erik* et *Crew* au format CIF 30 Hz. Pour les schémas AS2D+t et AS t, la taille de GOF choisie est 8. On rappelle que la longueur des arêtes choisie est de l'ordre de 8 pour la géométrie et de 16 pour le mouvement. Un flux est généré à l'encodage puis décodé à des débits compris dans $\{64, 128, 256, 512, 1024\}$ (kb/s). Pour SVC, nous choisissons également une taille de GOF égale à 8. Le mouvement est quantifié au 1/2 pixel. Les vidéos sont encodées de façon à générer trois couches de qualité correspondant à des « QP » égaux à 40, 34 et 28. Une couche de qualité est ajoutée pour

Akiyo correspondant à un QP de 50. Les PSNR affichés sont les PSNR des couches de qualité générées.

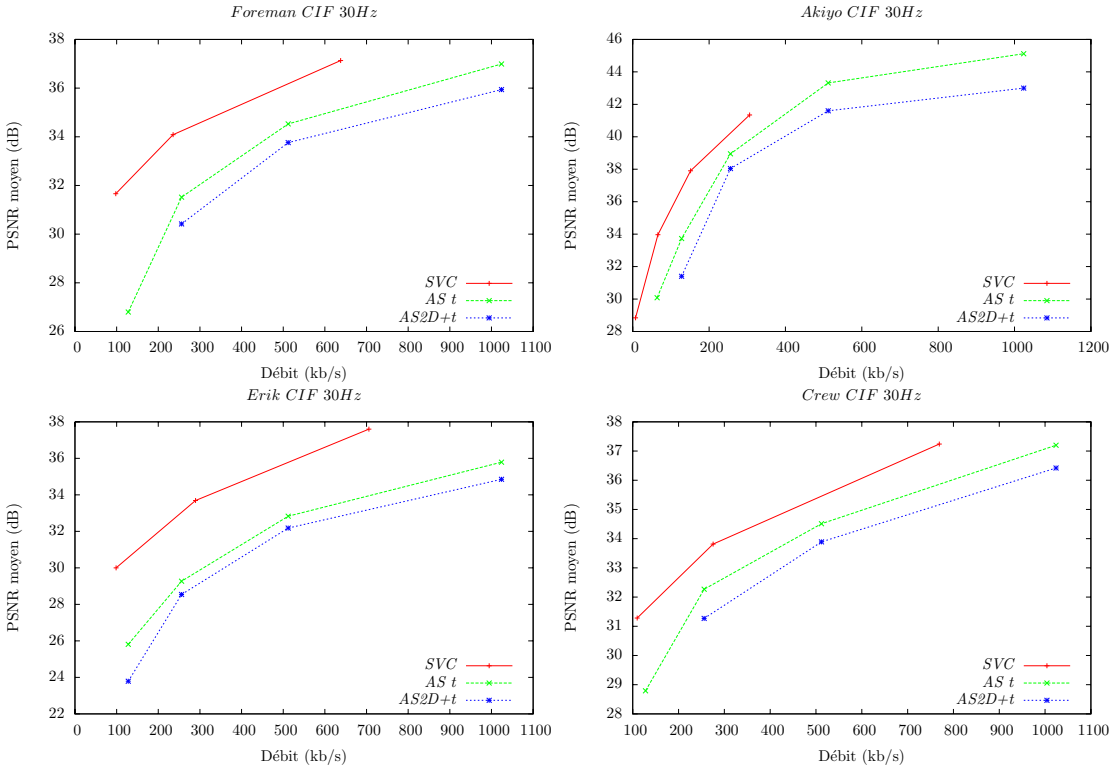


Fig. 5.14 : Courbes débit-distorsion obtenues pour quatre vidéo tests au format CIF 30 Hz.

La figure 5.14 montre les courbes débit-distorsion obtenues pour les trois codeurs. Ces courbes donnent le PSNR moyen sur les 100 premières images de chaque séquence (sauf pour la séquence *Erik* qui ne comporte que 50 images) pour chaque débit.

La première observation que nous faisons est que le standard SVC donne des performances meilleures à tous les débits que nos implémentations des schémas par analyse-synthèse. La qualité visuelle des séquences reconstruites est en accord avec ces courbes. Ces résultats peuvent être pondérés avec ceux fournis dans Cammas [Cam04b], où l'auteur montre que le schéma AS t apporte des gains visuels dans les bas débits par rapport au standard H.264/MPEG-4. La raison principale pour laquelle nos résultats sont moins bons est que notre estimateur de mouvement ainsi que l'encodage des positions ne sont pas optimisés comme ils le sont dans [Cam04b]. En particulier, une technique d'estimation hiérarchique pourrait être employée pour améliorer l'alignement temporel et un meilleur compromis entre la taille des mailles (c'est à dire le coût du maillage) et la qualité de l'alignement pourrait être recherché.

Une des questions que nous nous sommes posées en proposant le schéma AS2D+t est de savoir si la prise en compte de la géométrie lors de l'analyse allait apporter un gain par rapport au schéma AS t. Les courbes débit-distorsion nous montrent que ce

Tab. 5.1 : Coût des paramètres de compensation (kb/s).

video / codec	SVC	AS t	AS2D+t (mouvement + géométrie)
Akiyo	13,02	30,75	30,75 + 49,80
Foreman	42	68,61	68,61 + 59,52
Erik	29,45	57,8	57,8 + 38,58
Crew	46,89	73,65	73,65 + 56,58

n'est pas le cas. Les performances du schéma AS t sont meilleures que celles du schéma AS2D+t à tous les débits. L'explication principale de ces résultats est le coût de la géométrie. Le tableau 5.1 donne les coûts de l'information annexe pour chaque codeur : mouvement pour les schémas SVC et AS t, mouvement et géométrie pour le schéma AS2D+t. Comme on le voit, la géométrie occupe une part très importante du débit. Dans le cas d'*Akiyo*, le coût de la géométrie est même supérieur à celui du mouvement. Pour diminuer le coût de la géométrie, on pourrait augmenter la taille des mailles. Cependant, nous avons vu au chapitre précédent que ceci empêchait de s'adapter à des contenus variés. La figure 5.15 montre la première image de la séquence *Akiyo* reconstruite pour chaque codeur. Le débit cible choisi est celui atteint par SVC pour générer la troisième couche de qualité ($QP = 40$). Comme on le voit, la qualité de l'image reconstruite avec SVC est nettement au dessus de celle des images reconstruites avec les deux autres codeurs. Si on compare les images reconstruites avec les schémas par analyse-synthèse, on voit que les contours, particulièrement au niveau du buste de *Akiyo* sont mieux reconstruits avec le schéma AS2D+t. Cependant, l'image est dans l'ensemble un peu plus floue car la part de débit accordée au décodage des textures est moins importante. Ceci explique que le PSNR soit moins bon.

Les derniers travaux que nous avons menés dans cette thèse ont visé à améliorer la qualité de l'alignement temporel en appliquant le schéma AS t à d'autres modèles que le modèle par maillage déformable. Nous présentons la problématique et les résultats dans la section suivante.

5.3 Amélioration de la compensation temporelle

5.3.1 But de l'étude

Dans les schémas par analyse-synthèse proposés (AS2D+t etAS t), l'efficacité de l'estimation de mouvement détermine la précision de l'alignement temporel, la qualité de l'image basse fréquence I_{BF} (peu d'effets fantômes) et donc l'efficacité de l'estimation de géométrie. Précédemment, nous avons uniquement considéré un modèle de mouvement par maillage car il permet d'effectuer des compensations globales et *réversibles*. C'est également le modèle de mouvement utilisé dans les travaux antérieurs [WXCM99, Cam04b]. Cependant, nous remarquons qu'un tel modèle ne permet pas de représenter les discontinuités de mouvement dans les zones à occultations, ce

Originale | SVC (JSVM 8.9), PSNR = 39,95 dB

AS t , PSNR = 35,50 dB | AS2D+t , PSNR = 34,67 dB

Fig. 5.15 : Image originale au temps t_2 du premier GOF de la séquence *Akiyo CIF* $30Hz$ et images reconstruites à 151,5 kb/s par chaque codeur.

qui nuit sensiblement à la qualité des prédictions de I_{t_p} après compensation temporelle. Dans [Cam04b], Cammas propose de casser la structure du maillage en détectant des lignes de rupture dans la vidéo. La méthode proposée permet toujours de pouvoir reconstruire complètement les images à la synthèse, mais son efficacité dépend d'une segmentation préalable de la vidéo.

Le but de l'étude préliminaire présentée dans cette section est de casser la structure de façon automatique en appliquant l'analyse à d'autres modèles de mouvement que le modèle par maillage déformable. Les modèles utilisés et les algorithmes d'estimation associés ont été présentés au chapitre 3. Les modèles que nous avons testés sont le modèle translationnel par blocs [Ric03] noté BM (« Block Matching ») décrit page 82, le modèle par blocs recouvrants [OS94, SM00] noté $OBMC$ (« Overlapped Block Motion Compensation ») décrit page 83, ainsi que les modèles hybrides $SOBMC$ (« Switched OBMC ») et $SCGI$ [IM00] (« Switched Control Grid Interpolation ») décrits page 89. Deux résultats spécifiques nous intéressent en premier lieu : la qualité de l'analyse temporelle, c'est à dire la qualité des prédictions $\bar{I}_{t_p \to t}$ de I_{t_p}, et la qualité de la synthèse, c'est à dire des images reconstruites en bout de chaîne (sans codage).

Comme les nouveaux modèles étudiés ne sont pas réversibles, nous étudions l'impact des pixels multiplement et non connectés sur la qualité des images reconstruites.

5.3.2 Résultats d'analyse temporelle

Les résultats d'analyse présentés ci-dessous ont été obtenus en effectuant un suivi de mouvement avec les modèles cités précédemment. Pour ces tests, l'instant de projection choisi est le premier instant de chaque GOF ($t_p = t_0$). Nous travaillons sur les 6 premiers GOF de taille $N_G = 8$ de la séquence $Mobile\ CIF\ 30Hz$ qui présente des mouvements variés (translation du calendrier, rotation du ballon) et des zones à occultations. Chaque estimation de mouvement entre t_p et t a été réalisée avec les spécificités suivantes :

BM Le domaine à l'instant de projection est partitionné en blocs de taille 16×16 non recouvrants. Pour chaque bloc, un vecteur mouvement est déterminé par une recherche exhaustive dans une fenêtre de taille 16×16 autour du candidat d'origine (le vecteur mouvement estimé à $t - 1$). Les vecteurs sont estimés et encodés avec une précision au demi pixel. Des techniques permettant d'améliorer l'estimation ou de réduire sa complexité ont été citées au chapitre 3 page 91.

OBMC Le domaine à l'instant de projection est découpé en blocs recouvrants de taille 32×32 (figure 5.16, à droite). Deux blocs voisins se recouvrent à moitié de sorte que le nombre de vecteurs mouvement est égal à celui du BM. Dans nos tests, nous n'effectuons pas d'estimation spécifique prenant en compte le recouvrement des blocs. Nous réalisons simplement une recherche de type « Block Matching » indépendante pour chaque bloc de taille 32×32. Des techniques d'estimation plus spécifiques ont été citées au chapitre 3 page 92. Lors de la compensation en mouvement, un pixel de l'image à l'instant de projection est prédit par 4 valeurs venant de 4 blocs recouvrants. Ces valeurs sont pondérées pour établir la valeur prédite. Les poids sont obtenus en associant une

fenêtre de pondération à chaque bloc recouvrant. Dans nos travaux, nous avons choisi la fenêtre bilinéaire.

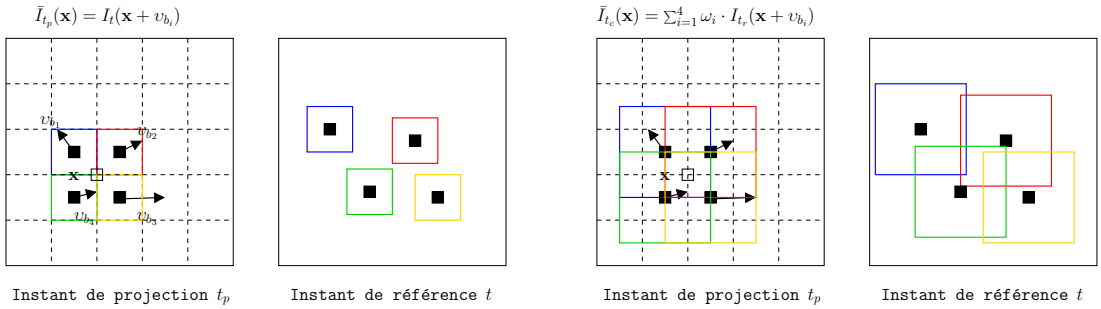


Fig. 5.16 : A gauche, modèle de mouvement par blocs non recouvrants (BM). A droite, modèle de mouvement par blocs recouvrants ($OBMC$).

SOBMC Pour générer un modèle $SOBMC$, nous utilisons la méthode simple qui est employée dans [IM00]. L'image I_{t_p} est découpée en blocs 16×16 et un « Block Matching » est effectué. Deux prédictions de l'image I_{t_p} sont ensuite calculées à l'aide des vecteurs mouvements : l'une en supposant comme lors de l'estimation que les blocs sont non recouvrants (BM) et l'autre en supposant que les blocs sont recouvrants ($OBMC$). Enfin, pour chaque bloc 16×16 dans l'image d'origine, nous comparons les erreurs quadratiques moyennes de prédiction données par le BM et l'$OBMC$ et affectons un label (valeur binaire) au bloc en fonction du résultat de la comparaison.

CGI-maillage déformable Pour le suivi de mouvement par maillage déformable, une grille uniforme est placée à l'instant de projection. Deux techniques d'estimation sont implémentées. La première est celle que nous avons utilisée à la section précédente qui s'appuie sur la technique de descente en gradient expliquée à l'annexe B. Pour être cohérent avec les paramètres des modèles précédents, les positions après suivi sont quantifiées au demi pixel avant de générer les images compensées. La seconde technique mise en œuvre est l'algorithme 1 décrit page 94. On rappelle que le principe de cet algorithme est de déplacer chaque nœud l'un après l'autre en fixant tous les autres de façon à minimiser l'erreur de prédiction sur le domaine d'influence du nœud uniquement. Les nœuds sont parcourus plusieurs fois jusqu'à ce qu'un minimum soit atteint pour chacun d'entre eux. *A chaque itération*, une recherche exhaustive est effectuée dans une fenêtre de dimension 6×6 centrée sur le vecteur mouvement optimal courant (comme prescrit par Sullivan et Baker dans [SB91]). *Sur l'ensemble des itérations*, la recherche est limitée à une fenêtre de 16×16 autour du candidat d'origine (le vecteur mouvement estimé à $t-1$). Les recherches sont réalisées avec une précision au demi pixel.

SCGI Dans [IM00], Ishwar et Moulin proposent d'estimer un modèle $SCGI$ en étendant l'algorithme précédent. Pour chaque nœud et chaque vecteur mouvement candidat, deux erreurs de prédiction sont calculées : l'une en considérant que la maille inférieure

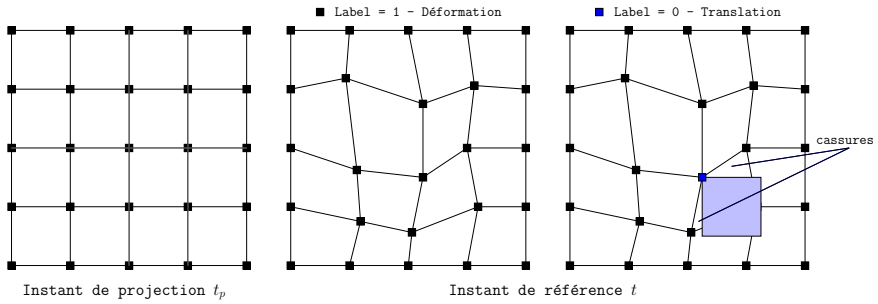


Fig. 5.17 : Maille déformable (*CGI*) et ajout d'un label (*SCGI*) pour accepter les cassures de connectivité.

droite du nœud suit un mouvement de translation indépendant des mailles voisines (cassure dans le maillage), l'autre en considérant qu'elle suit un mouvement de déformation bilinéaire en restant connectée à ses voisines (figure 5.17).

La figure 5.18 montre le résultat visuel des suivis de mouvement sur le premier GOF de 8 images de la séquence *Mobile*. Nous illustrons les paramètres de chaque modèle au dernier instant du GOF. Pour le modèle $SOBMC$, un carré jaune signifie que le BM est préféré à l'$OBMC$ lors de la compensation. Pour le modèle $SCGI$ un carré jaune signifie qu'il est préférable de casser la structure pour améliorer la prédiction de I_{t_p}. Comme on le voit, les différences les plus importantes entre les modèles apparaissent au niveau des zones à occultation, particulièrement celles autour du ballon. Dans la zone qui est découverte par le déplacement du ballon, la densité des blocs est moindre dans le cas du BM car cette zone ne prédit correctement aucun bloc de I_{t_p}. Dans le cas du maillage, cette zone produit des étirements de mailles car certaines mailles suivent le déplacement du ballon tandis que d'autres mailles proches suivent le déplacement du calendrier. Dans la zone recouverte par le ballon, les remarques sont inversées : nous observons une accumulation de blocs dans le cas du BM et une contraction des mailles dans le cas du CGI. Logiquement, dans le cas des modèles hybrides, ces zones à occultation provoquent des ruptures de continuité.

La figure 5.19 permet de comparer la qualité de l'alignement temporel sur l'instant de projection. Nous rappelons que plus l'alignement temporel est précis plus le GOF compensé en mouvement est adapté à une décomposition le long de l'axe temporel. Pour chacun des modèles, nous représentons le PSNR entre l'image à l'instant de projection et chacune de ses prédictions $\bar{I}_{t_p \to t}$. D'après ces courbes, on remarque tout d'abord que le modèle par maillage déformable (CGI) est celui qui donne les plus grandes variations de qualité : il donne les meilleures prédictions de I_{t_p} pour l'instant le plus proche de t_p mais aussi nettement les plus mauvaises pour les instants les plus éloignés. Lorsque le calcul du maillage se fait par optimisation locale, la qualité de la compensation en fin de GOF s'améliore parfois de façon importante. Si l'on compare maintenant le BM et l'$OBMC$, on remarque que le second modèle améliore la compensation sur 4 des 6 GOF traités tout en conservant la qualité du BM sur les 2 autres GOF. Le modèle hybride

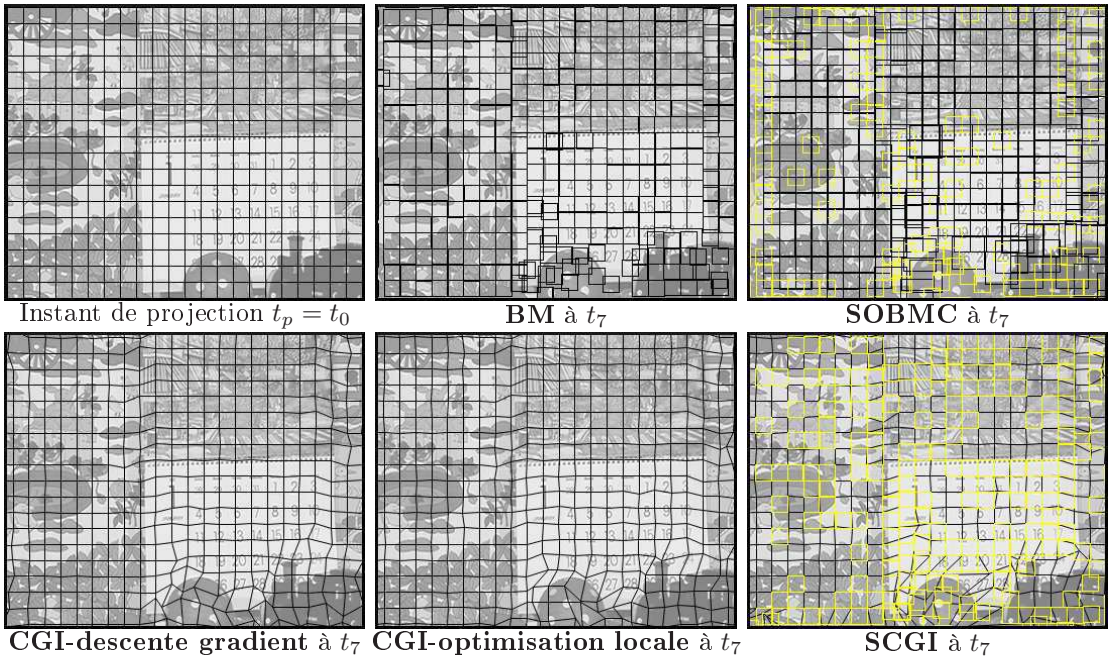


Fig. 5.18 : Mouvement entre $t_p = t_0$ et t_7 après un suivi sur toutes les images précédentes.

SOBMC apporte quant à lui un gain général par rapport à l'*OBMC*. Ceci n'est pas étonnant car ce modèle conserve pour chaque bloc dans I_{t_p} la meilleure prédiction entre celle donnée par le *BM* et celle donnée par l'*OBMC*. En observant maintenant les performances données par le *SCGI* on s'aperçoit que ce modèle améliore significativement la qualité de la compensation en fin de GOF par rapport au *CGI* obtenu avec optimisation locale. Puisque le label associé à chaque nœud donne le mouvement optimal de sa maille inférieure droite parmi une translation ou une déformation (voir figure 5.17), on aurait pu s'attendre à ce que ce modèle améliore les performances à la fois du *CGI* et du *BM* sur toutes les images. La différence entre le résultat et nos prévisions est due à l'algorithme de recherche employé pour le *SCGI* qui ne garantit pas d'aboutir au minimum local trouvé par le *CGI* ou le *BM* dans la fenêtre de recherche.

5.3.3 Résultats de synthèse

A la fin de l'analyse, nous avons obtenu pour chaque modèle un groupe d'images projetées sur le premier instant du GOF. Dans ce paragraphe, nous reconstruisons les images d'origine à l'aide de ces images compensées. Le principe général que nous avons adopté pour la reconstruction d'une image est de prendre indépendamment chaque bloc (maille) dans l'image compensée $\bar{I}_{t_p \to t}$ et de le (la) replacer à sa position et sa forme d'origine dans l'image I_t. Nous avons vu à la section précédente et au chapitre 4 que la qualité de ces images détermine les performances du codeur dans les hauts débits où le coût du mouvement devient marginal.

En utilisant les modèles *BM*, *OBMC*, *SOBMC* et *SCGI*, une nouvelle probléma-

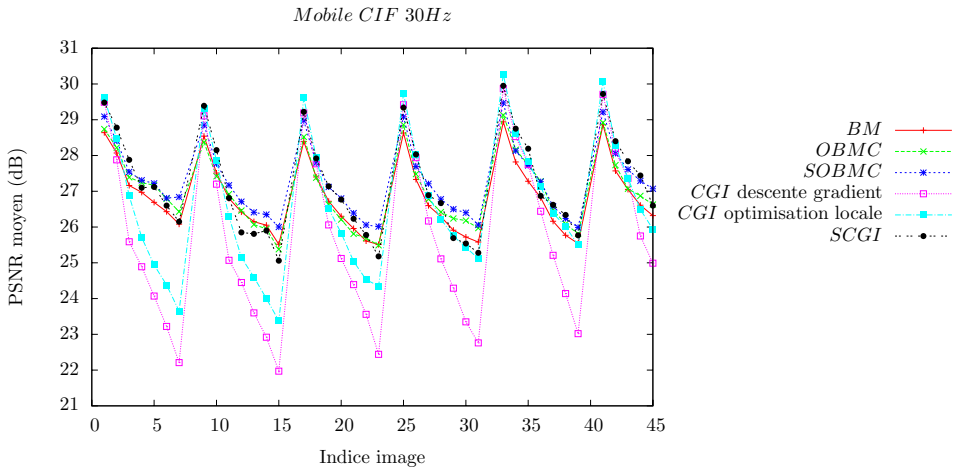


Fig. 5.19 : Qualité de l'alignement temporel en fonction des modèles de mouvement.

tique apparaît par rapport à la section précédente : ces modèles ne sont pas réversibles. En effet, comme expliqué au chapitre 3, des pixels non connectés ou multiplement connectés apparaissent lorsque l'on essaie d'inverser les correspondances avec ces modèles. Les pixels non connectés appartiennent majoritairement à des zones qui sont apparues par rapport à l'instant de projection. Les pixels multiplement connectés appartiennent majoritairement à des zones qui ont disparu par rapport à l'instant de projection.

Dans un premier temps, nous ne cherchons pas à reconstruire les pixels non connectés. Nous évaluons la qualité des zones reconstruites uniquement. Pour reconstruire les pixels connectés ou multiplement connectés, nous adoptons le principe suivant :

- à chaque intensité à l'intérieur d'un bloc « non recouvrant » dans le modèle BM ou d'une maille dans les modèles CGI et $SCGI$ est associé un poids de 1,
- à chaque intensité d'un bloc « recouvrant » dans le modèle $OBMC$ est associée la valeur d'une fonction de forme bilinéaire qui vaut 1 au centre du bloc et 0 sur les bords

Le cas du modèle $SOBMC$ est un peu plus complexe. En effet, lors de la compensation, un bloc recouvrant peut contribuer à prédire certains blocs voisins mais ne pas être utilisé pour prédire les autres blocs qui l'entourent. A la synthèse, nous devons faire un choix : soit considérer ce bloc comme recouvrant ou non recouvrant. Ici, nous choisissons de considérer tous les blocs comme recouvrants pour réduire au maximum le nombre de pixels non connectés.

Un pixel connecté ou multiplement connecté dans l'image à synthétiser peut donc être reconstruit à partir d'une ou plusieurs valeurs provenant d'une maille, d'un bloc non recouvrant ou d'un bloc recouvrant. A chacune de ces valeurs est associé un poids quelconque. Notons $\{I_k\}_k$ l'ensemble des valeurs candidates à la reconstruction d'un pixel et $\{\omega_k\}_k$ les poids qui leur sont associés. La valeur finale du pixel à reconstruire est obtenue en sommant les valeurs pondérées puis en normalisant le résultat. En notant $Combine$ la fonction utilisée pour combiner les valeurs candidates, ceci donne :

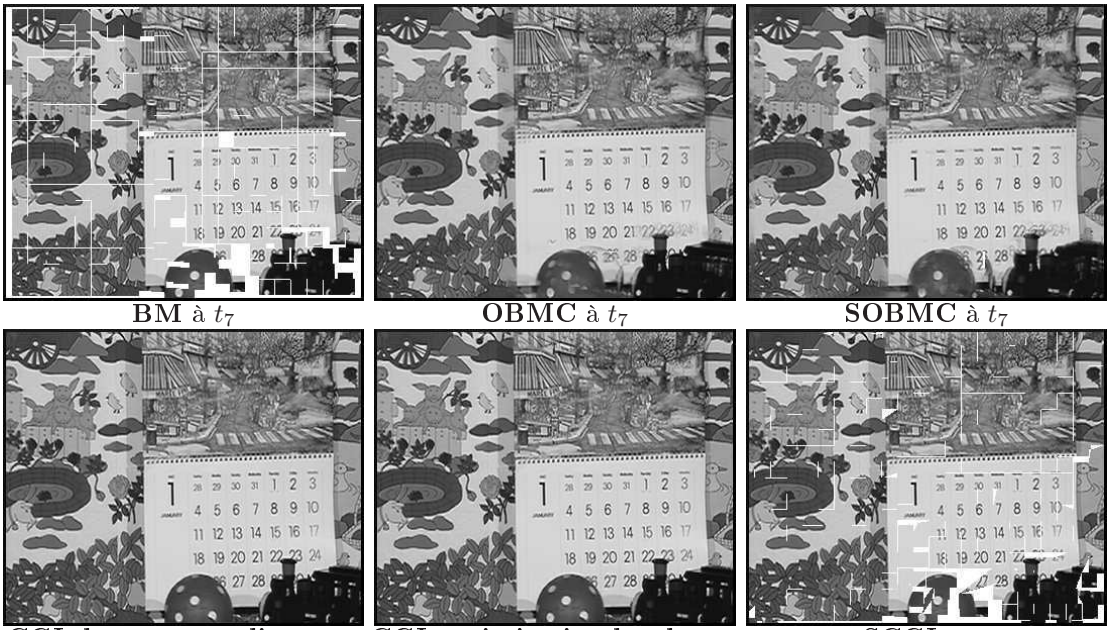


Fig. 5.20 : Image synthétisée à l'instant t_7 pour chaque modèle.

$$Combine(\{I_k\}_k, \{\omega_k\}_k) = \frac{1}{\sum_k \omega_k} \sum_k \omega_k \cdot I_k \tag{5.7}$$

Différentes fonctions *Combine* pourraient être choisies utilisant ou non les poids $\{\omega_k\}_k$ (valeur maximale, valeur moyenne, etc.).

Sur la figure 5.20, nous montrons l'image reconstruite à la fin du premier GOF avec chacun des six modèles étudiés. Le pourcentage de pixels non reconstruits au cours de la séquence pour les modèles BM, $OBMC$, $SOBMC$ et $SCGI$ est donné sur la figure 5.21. Sauf cas particuliers, on peut voir sur la figure 5.20 que les zones non reconstruites les plus larges sont des zones à occultation ainsi que l'intérieur du ballon dont le mouvement de rotation est difficilement capturé. Même si ces zones non reconstruites frappent l'œil, remarquons qu'elles occupent une part peu importante de l'image (autour de 10% *maximum* pour le BM, autour de 6% *maximum* pour le $SCGI$). En utilisant des blocs recouvrants, on parvient à reconstruire une valeur pour la quasi-totalité des pixels. Dans les zones qui ne sont pas des zones à occultations (tapisserie du fond et calendrier), la reconstruction obtenue avec les modèles $OBMC$ et $SOBMC$ est de bonne qualité. En revanche, dans les zones qui étaient cachées par le ballon et la locomotive à l'instant t_p, la reconstruction n'est pas satisfaisante. Dans le cas du $SCGI$, notons qu'il serait possible de réduire le nombre de zones non reconstruites en pénalisant les déconnections de mailles. Ceci aurait aussi pour effet de réduire l'entropie des labels.

La figure 5.22 donne les courbes de PSNR obtenues en comparant **les zones re-**

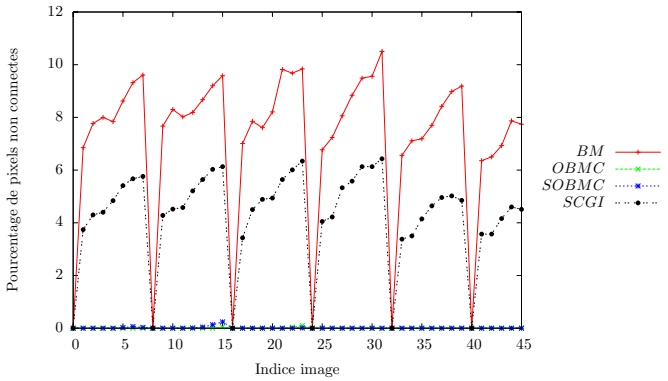


Fig. 5.21 : Pourcentage de pixels non reconstruits pour les modèles autorisant des déconnections de mailles (blocs).

construites de chaque image par rapport à leur contenu dans les images d'origine. On remarque tout d'abord que le maillage obtenu avec la technique d'optimisation locale permet une meilleure reconstruction des images que celui obtenu avec la descente en gradient. Nous avons vu qu'il permet également un meilleur alignement temporel. Ensuite, les modèles par blocs recouvrants *OBMC* et *SOBMC* sont ceux donnant les moins bons résultats particulièrement en fin de GOF où nous l'avons vu les zones découvertes sont mal reconstruites. Enfin, nous remarquons que les résultats de synthèse fournis par le *BM* et le *SCGI* sont semblables mais moins bonnes que celles du *CGI* obtenu avec optimisation locale.

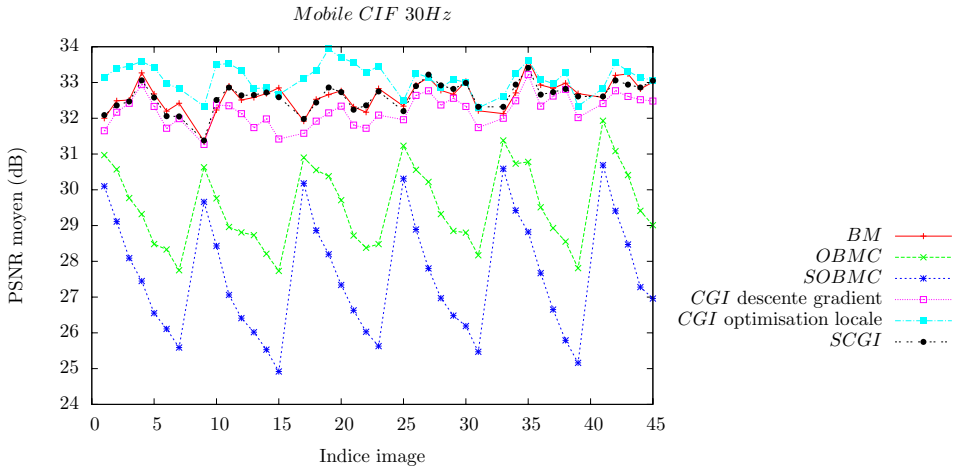


Fig. 5.22 : Qualité de la synthèse (sans codage) pour chaque modèle.

En résumé, nous avons vu que les modèles les mieux adaptés à notre problématique d'*analyse* sont les modèles par blocs recouvrants. Cependant, utiliser partout des blocs recouvrants lors de la synthèse aboutit à une mauvaise qualité dans les zones à occultation. Les modèles les mieux adaptés à notre problématique de *synthèse* sont les

maillages déformables. Cependant, ces modèles limitent l'alignement temporel lorsqu'on s'éloigne de l'instant de projection. Entre ces deux groupes, les modèles BM et $SCGI$ donnent des performances similaires. Le $SCGI$ permet néanmoins de réduire la proportion de pixels non connectés par rapport au BM. Ceci a bien sûr un prix en termes de coût de codage : le prix des labels. Dans le paragraphe suivant, nous présentons les résultats de codage obtenus avec les modèles BM, $SCGI$ et CGI (descente en gradient et optimisation locale).

5.3.4 Résultats de codage

Les résultats de codage présentés ici ont été obtenus en appliquant le schéma par analyse-synthèse AS t aux modèles BM, $SCGI$ et CGI. L'estimation géométrique proposée aux sections précédentes est désactivée. Le tableau 5.2 donne pour les quatre modèles étudiés le coût moyen de l'information de mouvement et des labels sur les six premiers GOF de la séquence *Mobile*. Le coût du maillage (CGI) est logiquement inférieur au coût du « Block Matching » car les déplacements sont contraints par la régularité du maillage. Dans le cas du $SCGI$, les labels ont un coût de $11,64$ kb/s. Notons qu'en moyenne 40% des mailles ont été déconnectées lors de chaque estimation de mouvement entre I_{t_p} et une image I_t. Comme nous l'avons noté précédemment, il serait possible de pénaliser la déconnection des mailles lors de l'estimation pour réduire l'entropie des labels. Dans notre cas, le modèle $SCGI$ coûte en moyenne un débit supérieur de 16 kb/s à ceux occupés par les modèles CGI. La question est de savoir si le compromis apporté par le $SCGI$ est meilleur que celui des modèles précédents.

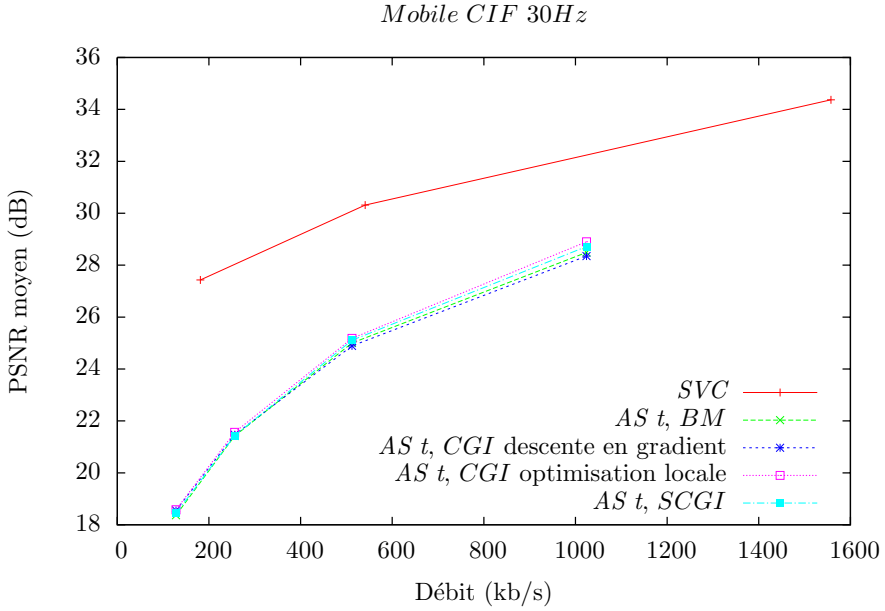


Fig. 5.23 : Courbes débit-distorsion obtenues en appliquant le schéma AS t avec différents modèles de mouvement. Pour le BM et le $SCGI$, le PSNR donné considère uniquement les zones reconstruites.

Tab. 5.2 : Coût des paramètres de compensation (kb/s).

modèle / information	mouvement	label
BM	74,04	
CGI (descente en gradient)	67,19	
CGI (optimisation locale)	67,00	
SCGI	72,02	11,64

La figure 5.23 donne les courbes débit-distorsion obtenues sur les six premiers GOF de la séquence *Mobile* avec les quatre modèles étudiés. Nous affichons également la courbe obtenue avec H.264/MPEG-4 SVC (JSVM 8.9). Comme nous pouvons l'observer, le modèle donnant le meilleur résultat est le maillage déformable obtenu avec optimisation locale. Le modèle *SCGI* n'apporte donc pas d'amélioration sur les zones reconstruites. De surcroît il génère des pixels non connectés dont la reconstruction est une problématique à part entière. Etant donnés les résultats obtenus, cette problématique ne sera pas traitée ici.

L'étude sur les modèles de mouvement appliqués au schéma AS t conclut les travaux que nous avons entrepris dans le cadre de cette thèse. Avant de présenter la conclusion générale de nos travaux, nous faisons un bilan de ce chapitre et proposons des perspectives qui feront l'objet d'études futures. En particulier, nous introduisons les premières briques d'une nouvelle représentation des séquences vidéo : les tubes de mouvement.

5.4 Bilan du chapitre

Méthode proposée. Dans ce chapitre, nous avons proposé un schéma de codage vidéo par analyse-synthèse dont le but est de tirer partie à la fois des corrélations temporelles (mouvement) d'un groupe d'images et des corrélations spatiales (géométrie) à l'intérieur de chaque image. Suivant le principe utilisé dans notre codeur d'images fixes, ce schéma s'appuie sur une adaptation du contenu spatio-temporel du groupe d'images à un filtrage selon les directions fixes temporelle-horizontale-verticale. Une première phase d'analyse permet de créer un GOF compensé en mouvement adapté à une décomposition temporelle. Une deuxième phase d'analyse calcule la géométrie de la basse fréquence temporelle du GOF compensé. En effet, si l'alignement temporel est efficace, cette géométrie est similaire à celle de toutes les images compensées. L'estimation de la géométrie se fait en suivant l'algorithme que nous avons proposé au chapitre précédent. La troisième phase de l'analyse permet finalement de générer un groupe de textures, versions déformées spatialement des images d'origine pour venir s'adapter à une décomposition temporelle-horizontale-verticale. Ce groupe de textures est transmis avec les paramètres de mouvement et la géométrie du GOF. Après réception et décodage, une étape de synthèse permet de reconstruire les images d'origine en inversant les déformations effectuées lors de l'analyse.

Distinction par rapport à l'art antérieur. Les travaux entrepris dans ce chapitre s'inscrivent dans la suite des travaux de Cammas [Cam04b] qui se plaçaient en rupture par rapport aux codeurs standards. Dans les standards, la décomposition temporelle est adaptée au signal après extraction du mouvement et chaque brique (transformée, quantification, codage) est optimisée pour une reconstruction de l'image pixel à pixel. Les schémas par analyse-synthèse proposés se distinguent principalement des standards par le fait que le signal d'entrée est *modifié* pour s'adapter à un noyau de décomposition fixe. Comme pour les schémas de codage par analyse-synthèse de modèles 3D [Gal02, Bal05], le but visé n'est pas la reconstruction des valeurs sur la base du pixel mais la qualité *visuelle* générale des images synthétisées.

L'élément qui distingue particulièrement notre schéma noté AS2D+t des codeurs précédents est la prise en compte de la géométrie des images tout en cherchant à limiter le coût de cette géométrie sur un GOF. De plus, nous avons vu qu'il est possible de représenter le mouvement et la géométrie par un même modèle : le maillage déformable. Dans ce cas, les algorithmes d'estimation du mouvement et de la géométrie sont très similaires.

Résultats. Bien qu'une seule géométrie soit transmise par GOF, les résultats obtenus en utilisant le maillage déformable pour représenter à la fois la géométrie et le mouvement ont montré que cette géométrie coûtait trop cher. En effet, afin de capturer des caractéristiques géométriques suffisamment variées, il est important de modéliser la géométrie avec un maillage dont les mailles ont une taille de l'ordre de 8×8. Or, une telle géométrie peut occuper une part de débit plus grande que le mouvement sur le GOF. L'adaptation spatiale ne permet pas de compenser ce coût. Comparé à un schéma de type AS t où seule l'analyse temporelle est effectuée, une meilleure reconstruction des contours peut être observée mais la qualité visuelle générale est moins bonne car les zones texturées sont moins bien reconstruites. Le standard de compression scalable H.264/MPEG-4 SVC donne des résultats meilleurs que nos implémentations des schémas par analyse-synthèse. Cependant, il est important de préciser que dans nos travaux le codage des informations de mouvement n'a pas été optimisé. Ceci conduit à un surcoût important par rapport à SVC qui code ses informations avec un codeur entropique puissant et optimisé (CABAC). Comme démontré précédemment [Cam04b], une estimation plus fine et un codage plus performant du mouvement dans le cas du schéma AS t permettent d'obtenir des résultats visuels meilleurs que ceux du standard dans les bas débits.

Le schéma AS2D+t que nous avons proposé est un schéma général qui peut être mis en œuvre avec d'autres modèles de géométrie et/ou de mouvement que le modèle par maillage déformable. Dans la dernière section du chapitre, nous avons par exemple appliqué le schéma avec des modèles autorisant des déconnections de mailles pour représenter le mouvement. Nos résultats montrent que ces modèles permettent un meilleur alignement temporel lors de la compensation en mouvement. Cependant, nous avons vu que ce gain ne se traduit pas par un meilleur compromis débit-distorsion en bout de chaîne. Le coût des labels nécessaires pour décider si une maille est déconnectée ou non est en partie responsable de ce résultat. De surcroît, autoriser la déconnection des mailles

fait apparaître des zones non connectées lors de la synthèse, dont la reconstruction reste un problème ouvert.

En ce qui concerne la géométrie, d'autres modélisations pourraient aussi être testées. En particulier, les schémas par déformation de blocs, comme la transformée en Bandelettes de première génération [PM05] ou la transformée de Taubman et Zakhor [TZ94b] pourrait aisément se substituer à notre estimation géométrique. Il serait intéressant de voir si de tels modèles offrent un meilleur compromis débit-distorsion dans le cadre du schéma AS2D+t.

Conclusion

Les travaux qui ont été développés dans ce manuscrit s'inscrivent dans la continuité des travaux menés par Cammas et Pateux [Cam04b]. L'objectif est de proposer et d'évaluer des *schémas en rupture* avec les standards de compression d'images fixes et de vidéos. Dans ce cadre, la ligne conductrice que nous avons suivie dans cette thèse consiste à **déformer les images pour adapter leur contenu à un noyau de décomposition fixe**.

Travail sur l'image fixe : problématique et contributions

Dans un contexte de compression d'images fixes, le standard actuel est JPEG2000. Ce standard s'appuie sur la décomposition en ondelettes séparables « classique », c'est-à-dire une décomposition 1D le long de l'axe horizontal suivi par une décomposition 1D le long de l'axe vertical. Au chapitre 1, nous avons mis en avant les limites d'une telle représentation : le noyau d'ondelette « classique » devient sous-optimal dès qu'il s'agit de représenter des caractéristiques géométriques (en particulier des contours courbes). Cette sous-optimalité se traduit par un phénomène de « ringing » près des contours lors d'une approximation de l'image à l'aide d'un nombre limité de coefficients. Au chapitre 2, nous avons présenté des outils antérieurs permettant une adaptativité au contenu des images, avec un focus particulier sur certaines ondelettes adaptatives. Parmi elles, la transformée en Bandelettes première génération s'appuie sur une segmentation de l'image en Quadtree. Chaque bloc est traité de façon indépendante : une détection de flux basée sur le gradient est effectuée puis le bloc est déformé en conséquence pour s'adapter au noyau d'ondelette classique. Tout d'abord, le fait de s'appuyer sur un a priori géométrique comme le gradient ne permet pas une modélisation du coût de codage des coefficients. Ensuite, traiter indépendamment chaque bloc du Quadtree nécessite une gestion particulière des bords et ne permet pas de ré-utiliser un codeur ondelettes comme JPEG2000.

Au chapitre 4, nous avons proposé un **nouveau schéma de codage d'images fixes par analyse-synthèse spatiales**. Ce schéma s'appuie sur un codeur d'images existant. Nous avons travaillé avec le codeur JPEG2000. La phase d'*analyse* a pour but de **déformer l'image pour adapter son contenu spatial au noyau d'ondelettes « classique »**. Pour modéliser la déformation de l'image, le *maillage déformable* est l'outil qui a été utilisé. Il permet d'effectuer une déformation de l'image globale, continue, et inversible. L'image déformée, que nous avons appelé *texture*, peut ainsi être

envoyée à un codeur JPEG2000 et bénéficier de ses propriétés (codage EBCOT, génération d'un flux « scalable », choix d'une zone d'intérêt...). L'estimation de la déformation ne s'appuie sur aucun a priori géométrique. Elle s'appuie sur **une modélisation du coût de description de la texture dans une base d'ondelettes**. Pour minimiser ce coût de description, une **technique d'optimisation** a été proposée qui **s'apparente fortement à une estimation de mouvement entre deux images**. Après analyse, encodage et décodage de la texture et du maillage, l'image d'origine peut être reconstruite en inversant la déformation effectuée à l'analyse.

Comme le maillage doit être transmis pour pouvoir synthétiser l'image en bout de chaîne, une question était de savoir si la réduction du coût de codage de l'image obtenue à l'analyse pouvait compenser le coût du maillage. Des premiers résultats ont montré qu'un maillage avec des mailles de taille de l'ordre de 8×8 coûtait trop cher à coder. Nous avons alors présenté des résultats en utilisant des mailles de taille 16×16 et en appliquant la méthode sur deux images (*Lena* et *Cameraman*) possédant un contenu géométrique simple. En comparaison avec JPEG2000, nous avons noté une **réduction significative du phénomène de rebonds** près des contours. Cependant, **deux limites principales** ont été mises en avant. Tout d'abord les **pertes numériques dues aux ré-échantillonnages** successifs lors de l'analyse puis de la synthèse amènent un **flou d'interpolation gênant surtout dans les zones texturées**. Ensuite, une taille de maille de l'ordre de 16×16 ne permet pas de capturer les structures géométriques plus complexes que l'on trouve par exemple dans l'image *Barbara* ou *Peppers*.

Pour remédier à ces difficultés, nous avons proposé plusieurs modifications au schéma de base. La **Quantification adaptative de la texture**, l'**Augmentation de la résolution de la texture** et la **Transmission d'une image de résidus** avaient pour but d'**améliorer la synthèse des zones texturées**. Si la quantification adaptative n'a pas porté ses fruits, les deux autres techniques ont permis de rehausser la qualité des images reconstruites. Pour pouvoir utiliser un maillage avec des mailles de l'ordre de 8×8, nous avons ensuite proposé **trois post-traitements à l'analyse pour limiter le coût de codage du maillage**. Le premier post-traitement consiste à « annuler » les déformations des mailles qui provoquent des pertes de qualité dans les zones texturées. Le critère utilisé pour quantifier ces pertes est l'index SSIM. « Annuler » les déformations signifie replacer les mailles sur leur carré d'origine. Le second post-traitement consiste à annuler les déformations non significatives des mailles en s'appuyant sur le jacobien local. Le troisième post-traitement consiste à créer un Quadtree avec les mailles carrées de façon à réduire le coût du maillage. En appliquant ces trois post-traitements et en comparant à nouveau les résultats de codage avec ceux fournis par JPEG2000, nous avons noté une **amélioration de la qualité visuelle générale des images pour des débits allant de** $0,3$ **à** $0,6$ **bpp**. En dessous, la part de débit occupée par le maillage est encore trop importante. Au-delà, les deux codeurs donnent des résultats visuels similaires.

Travail sur la vidéo : problématique et contributions

Dans le contexte de la vidéo, le standard de compression actuel est H.264/MPEG-4 AVC. Ce standard s'inscrit dans la lignée des standards précédents MPEG-x et H.26x. Il s'appuie sur le principe du codage prédictif. L'image courante est prédite à partir des images préalablement encodées par estimation puis compensation en mouvement, puis le résidu de prédiction est encodé et transmis. Au chapitre 3, nous avons présenté différents modèles de mouvement ainsi que des techniques utilisées classiquement pour estimer leurs paramètres. Nous avons ensuite montré qu'il existait différentes façons d'exploiter ce mouvement pour construire des schémas de codage. Le codage prédictif est une option mais il existe aussi des approches en marge. Nous avons en particulier décrit les approches par analyse-synthèse s'appuyant sur une modélisation 3D d'une séquence vidéo ou bien encore sur la création d'une mosaïque commune à plusieurs images. Le schéma par analyse-synthèse temporelles de Cammas et Pateux a ensuite été introduit. Le principe de ce schéma est de déformer les images d'un groupe d'images (GOF) pour adapter leur contenu temporel à un filtrage 1D « en ligne » le long de l'axe temporel.

D'une façon générale, nous avons observé que les schémas de codage vidéo antérieurs prennent en compte le mouvement entre les images mais ne prennent pas en compte la géométrie à l'intérieur de chaque image. La problématique principale est le coût de l'information annexe nécessaire pour modéliser cette géométrie. En particulier, utiliser un modèle géométrique pour chaque image dans le cadre de H.264/MPEG-4 s'avère en pratique trop coûteux, comme montré par les travaux de thèse de Robert [Rob08] réalisés en parallèle de notre étude.

Au chapitre 5, nous avons proposé un **nouveau schéma de codage de vidéos par analyse-synthèse spatio-temporelles**. Le but de ce schéma est de **déformer un groupe d'images pour l'adapter à une décomposition fixe le long de l'axe temporel puis le long des axes horizontaux et verticaux**. Il s'appuie sur celui de Cammas et Pateux. En effet, dans leur approche, chaque image du GOF compensé en mouvement possède un contenu géométrique que l'on peut exploiter. Puisque toutes ces images sont alignées sur un même instant de projection, leur contenu géométrique est cependant très similaire. En exploitant cette propriété, le nouveau schéma proposé permet de ne modéliser qu'**une seule géométrie par GOF.** L'**analyse spatio-temporelle** se déroule en **trois temps**. Dans un **premier temps** un suivi de mouvement est effectué sur le GOF puis chaque image est compensée en mouvement sur un même instant de projection. Cette première étape est l'analyse temporelle proposée par Cammas et Pateux. A l'issue de cette étape, le **GOF compensé en mouvement** est **adapté à une décomposition temporelle « en ligne »**. La **deuxième étape** consiste à **estimer une seule géométrie pour le GOF**. Cette géométrie est calculée sur une basse fréquence temporelle du GOF compensé en mouvement en utilisant l'analyse spatiale conçue pour l'image fixe. Enfin, la **troisième étape** permet de **générer un *groupe de textures* adapté à une décomposition 3D fixe** (temporelle, horizontale, verticale) en combinant compensation en mouvement et compensation en géométrie pour chaque image. Après décomposition temporelle avec une ondelette 1D, les images des sous-bandes sont envoyées au codeur JPEG2000 qui peut générer un flux « scalable ».

Après encodage, transmission et décodage des textures, des mouvements et de la géométrie, les images d'origine peuvent être synthétisées en inversant les déformations effectuées à l'analyse. Ceci nécessite l'utilisation d'un modèle de déformation inversible. En utilisant le **maillage déformable comme modèle pour la géométrie et les mouvements**, l'inversion est possible. De plus, nous avons montré que dans ce cas, l'**estimation de la géométrie et du mouvement s'effectuait via une technique similaire**. Nous avons alors évalué le schéma par analyse-synthèse spatio-temporelles noté AS2D+t en comparant ses performances avec un schéma par analyse-synthèse temporelles AS t et avec le standard de compression scalable H.264/MPEG-4 SVC. Sur des séquences CIF $30Hz$, nos résultats indiquent que **le standard donne des performances significativement meilleures que nos implémentations des schémas par analyse-synthèse**. Si l'on compare le schéma AS2D+t avec le schéma AS t, on s'aperçoit de plus que la prise en compte de la géométrie réduit les performances numériques. Des améliorations visuelles peuvent être observées au niveau des contours, mais la qualité générale des images est moins bonne. La raison de ces résultats est que **le coût de la géométrie représente une part trop importante du débit**.

La **dernière étude** que nous avons menée dans cette thèse avait pour objectif d'**améliorer l'analyse temporelle** décrite précédemment en choisissant d'autres modèles de mouvement que le maillage déformable qui ne permet pas de représenter les discontinuités de mouvement. Différents modèles (BM, OBMC, SOBM, CGI, SCGI) ont été testés. Les tests ont indiqué que **les modèles par blocs (BM, OBMC, SOBM) ainsi que le modèle hybride SCGI améliorent l'alignement temporel par rapport au maillage déformable (CGI)**. Malgré cela, nous n'avons **pas constaté d'amélioration significative des performances après codage et synthèse par rapport au maillage déformable**. Cette étude sur les modèles de mouvement a cependant fait naître une nouvelle idée de représentation d'une vidéo : les **tubes de mouvement**. Cette représentation offre des perspectives intéressantes que nous présentons ci-après.

Perspectives

Enseignements des travaux passés

Les approches par analyse-synthèse proposées dans cette thèse et dans celle de Cammas [Cam04b] pour le codage vidéo avaient pour but de rompre avec le schéma prédictif classique exploité et optimisé par les standards. Parallèlement à ces travaux, d'autres techniques adaptatives basées ondelettes, notamment le Barbell lifting [XXWL07], ont été implémentées. Bien que toutes ses technologies puissent être perfectionnées, on peut tenter de tirer quelques enseignements des travaux passés. Nous mettons en avant les points suivants :

Ondelettes et « scalabilité ». La propriété multi-résolutions de la transformée en ondelettes semblait fournir une réponse naturelle au problème de scalabilité spatiale. En pratique, on s'est aperçu que la scalabilité spatiale, notamment dans les techniques de type Barbell lifting, posait un certain nombre de difficultés en nécessitant par exemple le recours à des techniques dites 2D+t+2D [MT06]. En outre, les méthodes dites « bottom-up » des standards (voir page 109) semblent offrir une plus grande souplesse pour générer des formats d'images non dyadiques (type 4/3 ou 16/9) au décodage. Dans la direction temporelle, on s'est également aperçu que pour limiter les phénomènes de rebonds et effets fantômes lors d'une transformée ondelettes il était bénéfique de tronquer les mises à jour lors des étapes de lifting. Or, dans le cas d'une ondelette 5/3 ceci revient aux techniques « bottom-up » à base d'images B hiérarchiques mises en place dans les dernières normes.

Ondelettes et textures. Les capacités d'approximation d'une ondelette dépendent du contenu local représenté. Lorsque ce contenu comporte une information de contours, les techniques adaptatives présentées dans ce manuscrit offrent une qualité d'approximation parfois optimale. Cependant, dans le cas où le contenu comporte une zone texturée (damiers dans *Barbara*, plumes dans *Lena*, herbe dans *Cameraman*...) les performances d'une transformée en ondelettes (même adaptative) peuvent être moins bonnes que celles obtenues avec une DCT. Par ailleurs, des études comparatives entre H.264/MPEG-4 AVC et JPEG2000, ainsi que le développement par *Microsoft* du format *HD Photo*, montrent que des technologies basées blocs et DCT donnent des résultats aussi bons et parfois meilleurs que les technologies basées ondelettes. Les grandes forces des technologies basées blocs résident dans leurs algorithmes de quantification et

de codage entropique contextuel, dans leur capacité à adapter localement la taille des blocs au contenu des images, ainsi que dans leur capacité à limiter la dispersion des erreurs.

Maillage et compromis débit-distorsion. Le maillage comme l'ondelette est un outil multi-résolutions. Au cours de nos travaux, nous l'avons utilisé car il permet d'effectuer des compensations réversibles et peut être décodé de manière scalable. Cependant, la difficulté d'estimer les paramètres du maillage pour satisfaire un compromis débit-distorsion incite à revenir vers des modèles plus simples, en commençant par exemple par des modèles par blocs indépendants. La déconnection des blocs ou des mailles permet en outre de représenter plus fidèlement le mouvement dans les zones à occultation.

Toutes ces considérations, ainsi que l'étude menée au chapitre 5 section 5.3 sur les modèles de mouvement, nous ont amenés à poser les concepts généraux d'une nouvelle représentation du mouvement dans une vidéo. L'idée est d'introduire une structure de données élémentaire appelée *tube de mouvement*, qui permet de représenter une portion spatio-temporelle en mouvement sur un nombre quelconque d'images. Un tube de mouvement contient les caractéristiques (mouvement, déformation, mises à jour des intensités) relatives à une région de l'image qui évolue dans le temps. Un tube de mouvement peut naître ou mourir à tout instant selon les évolutions de la région (apparition, disparition, variation d'intensité). A un instant donné, une image est synthétisée en faisant un *rendu* à partir des informations portées par tous les tubes vivants. Chaque tube de mouvement peut être généré, suivi ou encodé indépendamment de ses voisins. Cette structure offre une grande souplesse face aux enjeux précédents. Ci-dessous, nous donnons des premiers éléments pour comprendre cette nouvelle structure et la façon dont elle peut être appliquée au codage. Ces éléments seront approfondis dans des travaux de thèse futurs.

Les tubes de mouvement : Vers une nouvelle représentation de la vidéo

Le tube de mouvement

Différents tubes de mouvement sont représentés sur la figure 5.24. Un tube de mouvement noté **T** est une structure de données qui possède trois types d'attributs :

1. **Cycle de vie.** Un tube de mouvement a un cycle de vie qui lui est propre. Il naît à un instant t_i et meurt à un instant t_f. Avant t_i, le tube n'existe pas. Après t_f, il n'existe plus.
2. **Position, forme initiale et déformations.** Un tube de mouvement est un volume 3D caractérisé par sa forme de départ (carré, quadrilatère quelconque, triangle...) et l'ensemble des déformations w_t qu'il subit à chaque instant. A un instant t donné, la région occupée par le tube dans l'image I_t est notée Ω_t. On a :

$$\Omega_t = w_{t-1} \circ \ldots \circ w_{t_i}(\Omega_{t_i}) \quad \forall t \in [t_i, t_f] \tag{5.8}$$

L'ensemble des formes possibles dépend du modèle choisi pour le suivi de mouvement. Si un modèle translationnel est choisi alors deux paramètres suffisent à décrire chaque déformation w_t. Si un modèle par blocs déformables est choisi alors 8 paramètres sont nécessaires pour un mouvement de déformation bilinéaire.

3. **Texture spatio-temporelle.** Un tube de mouvement est caractérisé par la texture qu'il contient à l'instant t_i.

   Optionnellement, un tube de mouvement $\mathbf{T}$ pourra contenir un ensemble de mises à jour permettant de rehausser à un instant t la qualité du bloc ou de la maille suivi(e). Cette option pourrait par exemple permettre de compenser des pertes de résolution.

Supposons maintenant qu'une séquence vidéo soit représentée par un ensemble de tubes de mouvement. Une image de la séquence à instant t peut alors être synthétisée en effectuant un rendu de tous les tubes existant à cet instant. On peut écrire la prédiction $\bar{I}_t$ de I_t comme :

$$\bar{I}_t = \bigcup_{\mathbf{T} \in \mathcal{L}_r} \mathbf{T} \tag{5.9}$$

où $\mathcal{L}_r$ est la liste des tubes disponibles à l'instant courant. Dans le cas où des tubes se chevauchent (instant t_2 dans la figure 5.24), un mécanisme de gestion des chevauchements peut être mis en place afin de définir quelle est la valeur reconstruite pour les pixels de l'image situés dans la zone de chevauchement. Cette problématique rejoint celle des pixels multiplement connectés et de l'élaboration d'une fonction de combinaison pertinente (moyenne, pondération spécifique à chaque tube, prise en compte du tube le plus récent uniquement...).

Application au codage

Établissement des tubes au sein du codeur. La façon de générer les tubes de mouvement est un problème ouvert. La méthode proposée ici s'appuie sur l'initialisation puis la mise à jour d'une liste de tubes de reférence $\mathcal{L}_r$. Par exemple, $\mathcal{L}_r$ peut être initialisée sur la toute première image de la séquence, notée I_0 en la découpant en blocs de taille constante ou variable (recouvrants ou non). Pour chaque bloc, un tube est créé. Supposons maintenant que l'on cherche à représenter I_1 en mettant à jour $\mathcal{L}_r$.

Pour chaque tube $\mathbf{T} \in \mathcal{L}_r$, nous proposons de mettre en place l'algorithme en 3 points suivant :

1. **Suivi.** On effectue le *suivi* de $\mathbf{T}$ de $t = 0$ à $t = 1$. Par exemple, en cherchant les vecteurs déplacements qui mèneraient à la meilleure prédiction d'une région (a priori inconnue) de I_1. Ceci requiert donc la mise en place d'une estimation de mouvement « forward » contrairement à toutes les méthodes citées précédemment qui implémentent des estimations de mouvement « backward ».

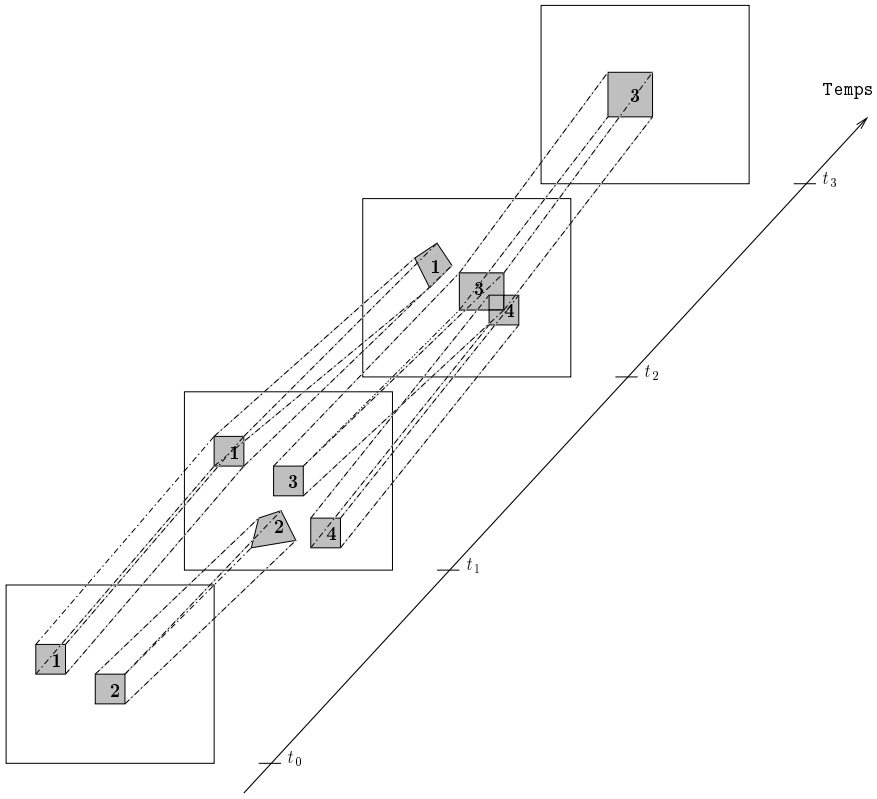


Fig. 5.24 : Cycle de vie et structure de différents tubes de mouvement.

2. **Décision.** Si des zones disparaissent ou changent sensiblement entre $t = 0$ et $t = 1$, certains tubes peuvent donner de mauvaises prédictions de I_1. Dans un deuxième temps, il est donc important de décider si $\mathbf{T}$ sera effectivement utilisé pour représenter I_1. Ceci pourra se faire en utilisant une technique de seuillage sur l'erreur de prédiction obtenue pour $\mathbf{T}$.
3. **Mise à jour.** Si on décide d'utiliser $\mathbf{T}$, une mise à jour de sa texture pourra être calculée pour l'instant $t = 1$ selon la qualité de la prédiction qu'il fournit.

A l'issue des 3 points précédents, l'ensemble des tubes de référence choisis et mis à jour peut ne pas suffire pour synthétiser l'image I_1 sur tout son domaine. Par exemple, des zones qui apparaissent ne peuvent être prédites correctement par aucun tube de la liste. Dans un second temps, nous proposons alors de découper I_1 en blocs de taille constante ou variable. Pour chaque bloc b, on décide s'il peut être reconstruit par la liste des tubes choisis précédemment. Si ce n'est pas le cas, alors on décide d'initialiser un nouveau tube à partir des données de b. Le nouveau tube vient s'ajouter $\mathcal{L}_r$.

Dans la description précédente, nous avons supposé que $\mathcal{L}_r$ était initialisée à l'instant $t = 0$ et que la première image à coder était I_1. Dans la pratique, on peut imaginer une structure de codage bien plus générale. Après avoir encodé et décodé une liste d'images à instants quelconques (par exemple dans l'ordre d'une structure de type $IBBP$ ou autre), on dispose d'une liste de tubes $\mathcal{L}_r$. On peut ainsi choisir d'encoder une nouvelle image à un instant quelconque en suivant les étapes précédentes.

Codage des tubes. Pour un instant t quelconque, plusieurs informations doivent être encodées afin de synthétiser l'image I_t au décodage. En particulier, pour chaque tube $\mathbf{T}$ dans la liste de référence, il faut coder une information binaire indiquant si t est utilisé ou non pour synthétiser I_t. Si $\mathbf{T}$ est utilisé, il faut ensuite coder l'information de mise à jour de la texture du tube et de la position. Il faut ensuite coder une information indiquant si de nouveaux tubes sont créés pour l'image en cours et pour chaque tube créé coder ses valeurs de texture et sa position d'origine. Enfin, on pourra coder d'autres informations permettant la mise à jour de $\mathcal{L}_r$, par exemple la suppression de certains tubes de mouvement.

Plusieurs façons d'encoder la texture (texture d'origine ou mise à jour) d'un tube pourraient être mises en concurrence. Si on modélise le mouvement par des blocs non déformables, on pourrait par exemple ré-utiliser certains modes de codage des standards. En fonction, du contenu d'un tube (contours, zones texturées, zones homogènes), on pourrait aussi choisir un noyau de représentation adapté (ondelettes adaptatives, DCT, ondelettes). Beaucoup de possibilités sont à étudier. La structure de décomposition temporelle est aussi à définir.

Décodage. Le décodage des informations suit le processus inverse de l'encodage. Pour reconstruire une image à un instant t quelconque, l'important est de connaître l'ensemble des tubes existants et utilisés à cet instant, et leur état. La reconstruction de I_t se fait ensuite en *combinant* les prédictions issues des différents tubes. Chaque tube $\mathbf{T}$ utilisé définit en effet une prédiction $\bar{\Omega}_{\mathbf{T}}(x, y, t)$ à l'instant t, où Ω est une région de l'image

d'origine (délimitée par un bloc ou une maille). La prédiction peut être obtenue par compensation en mouvement de la texture d'origine du tube et de ses diverses mises à jour éventuelles. La synthèse de l'image I_t peut alors s'écrire :

$$I_t(x,y) = \sum_{\mathbf{T} \in \mathcal{L}_r{}^*} \omega_{\mathbf{T}}(x,y,t) \cdot \bar{\Omega}_{\mathbf{T}}(x,y,t) \tag{5.10}$$

où $\mathcal{L}_r{}^*$ est la liste des tubes effectivement utilisés. Les poids $\omega_{\mathbf{T}}$ sont introduits afin de prendre en compte les pondérations à appliquer lors de prédictions multiples. Le choix de ces poids restent à déterminer. Un choix simple serait de considérer la moyenne de toutes les prédictions en un pixel, mais des poids spécifiques à chaque tube pourraient également être définis.

Atouts par rapport à nos travaux précédents

Dans les schémas AS t et AS2D+t qui ont été proposés, le suivi de mouvement est un suivi de type « Backward » et le critère d'optimisation est l'erreur de prédiction de l'image de projection I_t. En opérant ainsi, nous avons privilégié l'analyse (l'alignement temporel) au dépend de la synthèse. Dans le cas des tubes de mouvement, une image quelconque I_t est prédite à l'aide d'une liste de tubes courante en mettant à jour la position et la forme de ces tubes. On se trouve alors dans une estimation de type « Forward » et le critère d'optimisation est l'erreur de prédiction de l'image courante. Ce type d'optimisation permettra de mieux contrôler la distorsion qui sera réellement perçue par l'utilisateur en bout de chaîne.

Les études menées sur les précédents schémas par analyse-synthèse s'appuyaient essentiellement sur le maillage déformable régulier comme modèle de mouvement, afin d'assurer la réversibilité des déformations. Or, comme nous l'avons indiqué, ce modèle ne permet pas de représenter les discontinuités de mouvement dans les zones à occultation. Des travaux comme les *lignes de rupture* ont été proposés pour « casser » la régularité de la structure dans ces zones. Cependant la détection de ces zones, le maintien et le codage d'une structure à connectivité irrégulière ainsi que le codage des textures ne sont pas triviaux. Les tubes de mouvement adoptent une approche opposée en s'appuyant sur des structures élémentaires *indépendantes*. Chaque tube de mouvement peut évoluer indépendamment de ses voisins et donc modéliser des mouvements discontinus. La possibilité de connecter des tubes de mouvement voisins pour réduire le coût des déplacements dans les zones où le mouvement est continu reste envisageable. De plus, lorsqu'une zone dans l'image courante I_t n'est prédite correctement par aucun tube (zone qui apparaît), la solution sera de faire naître un nouveau tube dont la forme de départ couvre la zone.

L'indépendance des tubes de mouvement permet aussi d'envisager de nouvelles options par rapport aux schémas précédents. Par exemple, le choix du noyau de représentation spatial (DCT, ondelettes, ondelettes adaptatives) pourra différer selon le contenu du tube. La scalabilité temporelle pourra être mise en œuvre en adoptant un ordre de parcours des images similaire à celui des standards. L'accès aléatoire à une image quelconque dans la séquence peut aussi être envisagée. Cette image serait créée en effectuant

un rendu à l'aide des tubes de mouvement décodés disponibles, en mettant en place un traitement spécifique pour les zones non reconstruites (par un exemple un algorithme de « inpainting »).

Comme nous l'avons remarqué plus haut, une zone de l'image couramment traitée pourra être prédite par plusieurs tubes de mouvement qui se chevauchent. Ces chevauchements peuvent être vus comme un désavantage car ils génèrent de la redondance. Cependant, ils pourront aussi constituer un nouvel atout par rapport aux schémas antérieurs. En effet, introduire des redondances permet de transmettre des descriptions multiples d'un même objet qui, si elles sont quantifiées et combinées de façon pertinente, peuvent aboutir à un meilleur compromis débit-distorsion qu'une seule description à échantillonnage critique. En outre, transmettre des descriptions multiples pourra favoriser de nouvelles fonctionnalités comme la super-résolution spatiale et temporelle.

Questions ouvertes

Les tubes de mouvement constituent une nouvelle façon de représenter une vidéo. Nous avons posé les concepts généraux d'un codage à base de tubes de mouvement mais cette nouvelle représentation soulève un grand nombre de questions. Nous soulignons ci-dessous quelques points importants :

Suivi de mouvement forward. Comme nous l'avons mentionné précédemment, la mise à jour des positions d'un tube à un instant t se fait en effectuant une estimation de mouvement « forward » pour prédire une région de l'image I_t. Cette estimation est plus complexe qu'une estimation « backward » car elle nécessite une inversion de mouvement si l'on veut connaître l'erreur de prédiction dans l'image courante. Dans le cas d'un mouvement par blocs translationnel, l'estimation conserve une complexité similaire. Cependant, si l'on souhaite faire un suivi avec des blocs déformables, la complexité peut exploser : pour chaque déplacement de nœud testé, l'ensemble des pixels prédits changent et la déformation inverse doit être re-calculée. Dans le cas d'une déformation bilinéaire, le calcul de la déformation a une complexité non négligable, d'autant plus si les correspondances se font entre deux quadrilatères quelconques. Dans le cas d'un modèle *CGI* ou *SCGI*, le déplacement d'un nœud influencera la prédiction de plusieurs tubes voisins, ce qui rendra complexe la mise en place d'une optimisation débit-distorsion.

Contrôle de la liste. Les zones qui apparaissent à l'instant t ne peuvent pas être prédites correctement avec les tubes disponibles. D'autre part, l'utilisation d'un modèle par blocs indépendants peut introduire des discontinuités multiples dans le champ de mouvement, même lorsque le champ réel est continu. Ces discontinuités font apparaître des zones non prédites lors du suivi comme illustré à la fin du chapitre précédent. Dans le schéma proposé, toutes ces zones non prédites vont engendrer la création de nouveaux tubes. Une question majeure est le contrôle de la taille de la liste de référence pour ne pas faire exploser les ressources mémoire requises. En outre, le nombre de tubes utilisés à un instant t influe directement sur la redondance introduite. La question est de savoir comment gérer cette redondance (par exemple en la considérant comme une description

multiple) pour obtenir un compromis débit-distorsion satisfaisant. La question de la mort d'un tube doit également être résolue. Comment décider de supprimer un tube de la liste de référence ?

Gestion des chevauchements. A un instant t plusieurs tubes peuvent se chevaucher et la façon de combiner les valeurs candidates à la reconstruction d'un pixel reste un problème ouvert. Lors du suivi de mouvement, les chevauchements ne seront pas connus par avance et donc la distorsion réelle de l'image synthétisée sera difficilement exprimable si l'on souhaite prendre en compte toutes les contributions. D'autre part, à la synthèse les redondances introduites ne seront a priori pas homogènes sur l'ensemble de l'image. Si l'on souhaite utiliser des techniques de descriptions multiples, celles-ci devront être adaptées à l'information disponible. Un mécanisme spécial pourrait être nécessaire pour assurer une qualité homogène sur l'ensemble du domaine image.

Les trois points précédents ne sont que des exemples des problématiques soulevées par la représentation en tubes de mouvement. Des travaux de thèse futurs seront menés pour évaluer le potentiel du schéma de codage proposé en termes de compression et pour préciser ses propriétés et son champ d'applications. Comme pour tous les schémas d'analyse-synthèse, la métrique d'évaluation de qualité jouera aussi un rôle important dans la mise en valeur des résultats.

Annexe A

Création d'un maillage par intégration de lignes de flux géométrique

Cette annexe décrit la première piste que nous avons explorée dans l'objectif de créer un maillage quadrangulaire adapté à la géométrie d'une image. Rappelons que dans nos travaux ce maillage modélise une déformation dont le but est d'adapter le contenu de l'image à une décomposition en ondelettes séparables « standard », c'est-à-dire horizontale-verticale. Au chapitre 4 (page 118), nous avons souligné qu'adapter un contour à l'ondelette standard pouvait se décliner en trois points :

1. Orienter sa direction de régularité le long de l'axe horizontal ou vertical,
2. Contracter le contour le long de sa direction de régularité,
3. Étirer le contour dans sa direction orthogonale.

Pour atteindre ces objectifs, notre idée de départ était de construire des mailles quadrangulaires par intégration de lignes de flux en s'appuyant sur une caractéristique locale comme le gradient. Si l'on considère un contour quelconque, les lignes de flux permettront d'atteindre les objectifs précédents si :

1. Un premier réseau de lignes parallèles au contour est construit,
2. Un second réseau de lignes orthogonales au contour est construit,
3. Dans chaque réseau, l'espacement entre les lignes est proportionnel à la régularité locale.

Cette idée de départ nous est venue du domaine de la 3D, et plus particulièrement d'un article de Alliez et al. [ACSD+03]. Dans cet article, les auteurs proposent de remailler une surface 3D à l'aide de mailles majoritairement quadrangulaires adaptées à la géométrie de la surface. Le principe suivi dans cet article est illustré sur la figure A.1. Les auteurs partent d'une surface 3D discrétisée à l'aide d'un maillage triangulaire dont la densité des nœuds n'est pas adaptée à la géométrie locale. Un tenseur de courbure est alors estimé en chaque sommet du maillage. Ce tenseur comprend une *direction*

principale, la direction de courbure maximale, une *direction secondaire*, la direction de courbure minimale, ainsi que les valeurs mesurant la courbure dans ces deux directions. Les auteurs proposent ensuite d'intégrer deux ensembles de lignes à partir de ce champ : un ensemble de lignes de courbure maximale et un ensemble de lignes de courbure minimale. En fusionnant ces deux réseaux et en leur appliquant un post-traitement, les auteurs parviennent à dégager une structure maillée à dominance quadrangulaire. L'espacement entre deux lignes de courbure dans un réseau dépend localement de la valeur de cette courbure. Ainsi, les arêtes et mailles résultantes permettent une description compacte et fidèle de la géométrie de la surface.

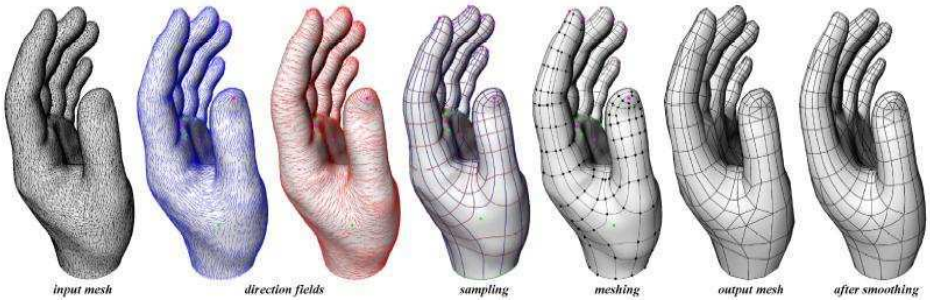


Fig. A.1 : Principe du remaillage anisotropique proposé dans [ACSD+03].

Notre idée est de suivre un raisonnement similaire pour calculer un maillage quadrangulaire adapté à la géométrie d'une image fixe en la considérant comme une surface 3D où la $3^{ème}$ coordonnée est la luminance. Dans la première section de cette annexe, nous proposons un court état de l'art sur la génération de lignes de flux à partir d'un champ vectoriel 2D. Ensuite, nous montrons comment Alliez et al. ont utilisé les techniques existantes pour mener à bien leur remaillage anisotropique. Enfin, nous décrivons notre travail et les conclusions auxquelles il nous a amenées.

A.1 Etat de l'art sur la génération de lignes de flux

Dans [Meb04], Mebarki fait un tour d'horizon des méthodes permettant le placement de lignes de courant à partir d'un champ de vecteurs 2D. Le problème est illustré sur la figure A.2 : étant donné un champ de vecteurs 2D, comment générer des lignes de courant qui permettent d'interpréter ce champ avec fidélité ? La motivation principale derrière la construction de telles lignes est la visualisation de champs de vecteurs stationnaires 2D, mais certaines approches ont aussi été appliquées au rendu non-photoréaliste ainsi qu'au remaillage de surfaces comme nous le verrons dans la section suivante.

Un bon placement de lignes de courant doit satisfaire 2 types de critères :

Validité

1. Conformité : toute ligne de courant doit être tangente au champ en chacun de ses points.

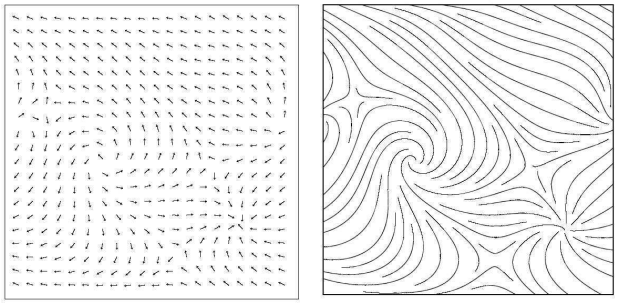


Fig. A.2 : Un champ de vecteurs initial et le placement de lignes de courant engendré par l'algorithme de Mebarki. D'après [Meb04].

2. Saturation du domaine : le réseau de lignes construit doit couvrir tout le domaine.

Qualité

1. Longueur des lignes de courant : les longues lignes de courant doivent être privilégiées car les terminaisons de lignes sont perçues comme des singularités, distrayant le regard de l'observateur.
2. Densité contrôlable : La distance séparatrice entre deux lignes de courant doit approcher au mieux une densité idéale soit spécifiée globalement par l'utilisateur, soit déterminée localement par un champ de densités.

Une ligne de courant est modélisée par une liste de points calculés par intégrations successives. L'intégration commence à partir d'un germe et se termine lorsque le point intégré se trouve hors du domaine de visualisation, ou lorsque la distance qui le sépare des autres points est inférieure à un seuil. Les 2 passages critiques de l'algorithme sont ainsi la détermination de la position du prochain germe, et la détermination du seuil (distance de séparation). Deux techniques « naïves » peuvent être avancées pour le choix des germes :

Placement sur grille régulière : ce choix laisse apparaître des motifs sur le placement final des lignes de courant.

Placement aléatoire : ce choix ne garantit pas la saturation du domaine.

Dans [JL97], Jobard et Lefer proposent de semer les points germes de manière adaptative. Les germes sont en effet semés de part et d'autre d'une ligne de courant en cours d'intégration. Cette méthode produit des placement de qualité et est largement utilisée. Cependant, elle a tendance à laisser de petites espaces non comblés et à créer des lignes de flux de petites tailles car les points germes sont placés à proximité des lignes de flux déjà générées.

Verma et al. [VKP00] choisissent quant à eux de privilégier la topologie du flux en démarrant l'intégration des lignes de courant dans le voisinage des points critiques où les vecteurs du champ partent ou arrivent de plusieurs directions. Cependant, ils ne proposent pas de contrôle sur la densité et la longueur des lignes, et font un choix aléatoire pour les germes après avoir saturé le voisinage des points critiques.

Pour parvenir à un placement bien réparti sur le domaine en privilégiant des longues lignes de courant, Mebarki [Meb04, MAD05] propose de semer les germes dans les plus grand espaces encore non comblés. Pour ce faire, il utilise une triangulation de Delaunay comme structure de données pour relier tous les points des lignes de flux déjà intégrées. Ceci lui permet de mener à bien toutes les requêtes de proximité en un temps réduit. Il propose par ailleurs un certain nombre d'optimisations qui permettent d'accélérer l'algorithme tout en limitant l'impact sur le résultat final.

A.2 Application au remaillage de surfaces

Comme nous l'avons noté, le placement de lignes de courant offre d'autres applications que la simple visualisation de flux. Dans [ACSD+03], Alliez et al. utilisent ces techniques d'intégration pour créer deux réseaux indépendants de lignes de courbure leur permettant de remailler des surfaces 3D avec un maillage majoritairement quadrangulaire. Leur travail se base sur la calcul d'un tenseur de courbure en chaque sommet du maillage d'origine. Un tenseur de courbure permet de définir localement deux directions principales, les directions de courbure maximale et minimale, ainsi que les valeurs de ces courbures tridimensionnelles. Pour faciliter l'intégration des lignes de courbure, les auteurs projettent le champ de tenseurs dans un plan en utilisant une paramétrisation dite *conforme* [FH05]. Ceci aboutit à deux champs vectoriels 2D $\{\vec{\gamma}_{min}\}$ et $\{\vec{\gamma}_{max}\}$, ainsi que deux champs scalaires correspondant aux courbures locales associées $\{k_{min}\}$ et $\{k_{max}\}$. Un lissage du tenseur dans l'espace de paramétrisation peut être effectué en fonction du détail souhaité pour le maillage final. Alliez et al. détectent aussi les points critiques (dégénérés) correspondant aux zones isotropiques (sphériques, plates) où aucune direction n'est privilégiée. Ces points sont appelé *ombilics*. De plus, l'algorithme prend en compte les lignes de démarcations (ou « features ») d'une surface : ces lignes sont des frontières à forte discontinuité le long desquelles le lissage est prohibé et au travers desquelles aucune lignes de courbure ne doit passer. Les réseaux de lignes de courbure maximale et minimale sont construits indépendamment, une intégration Runge-Kutta d'ordre 4 à pas adaptatif étant utilisée pour une ligne donnée.

La méthode proposée par Alliez et al. pour le choix des germes est une version hybride de [JL97] et [VKP00]. En effet, comme dans [VKP00], les premières germes choisies pour débuter la construction d'un réseau sont les ombilics, classés selon leur valeur de courbure associée. Puis, à chaque nouveau pas d'intégration, une paire de germes, placées orthogonalement à la ligne courante et à une distance idéale, est ajoutée à la liste des germes candidates comme dans [JL97]. L'intégration d'une ligne s'achève dans les quatre cas suivants :

1. La ligne atteint un ombilic,
2. La ligne revient à proximité de son point de départ,
3. La ligne croise une « feature » ou la frontière du domaine,
4. La ligne s'approche trop d'une ligne existante.

Les notions de *distance optimale* et de *proximité* dépendent de la densité de lignes souhaitée. La plupart des méthodes de l'état de l'art présentées plus haut ne gèrent qu'une densité globale. Alliez et al. se fixent une contrainte plus exigeante, puisque en chaque point du domaine la densité de lignes intégrées doit refléter une valeur de courbure locale. Etant donnée une valeur de courbure K, Alliez et al. expriment la distance locale idéale entre deux lignes du réseau par la relation suivante :

$$d(K) = 2\sqrt{\epsilon\left(\frac{2}{|K|} - \epsilon\right)} \tag{A.1}$$

où ϵ représente l'erreur d'approximation locale tolérée entre le maillage 3D résultant et la surface d'origine. Étant donnée la distance locale idéale $d(K)$, les germes sont alors placés dans une queue de priorité en fonction de la différence entre cette distance idéale requise et la distance réelle aux lignes déjà intégrées. Une triangulation de Delaunay contrainte est utilisée pour mener à bien les requêtes de proximité. Hormis ϵ, Alliez et al. définissent un second paramètre de contrôle ρ correspondant au degré d'anisotropie qu'ils souhaitent intégrer au maillage final. Les distances idéales entre deux lignes de courbure maximale d_{max} et deux lignes de courbure minimale d_{min} (équation (A.1)) en un point donné du domaine s'expriment alors en fonction des deux valeurs de courbure en ce point k_{min} et k_{max} :

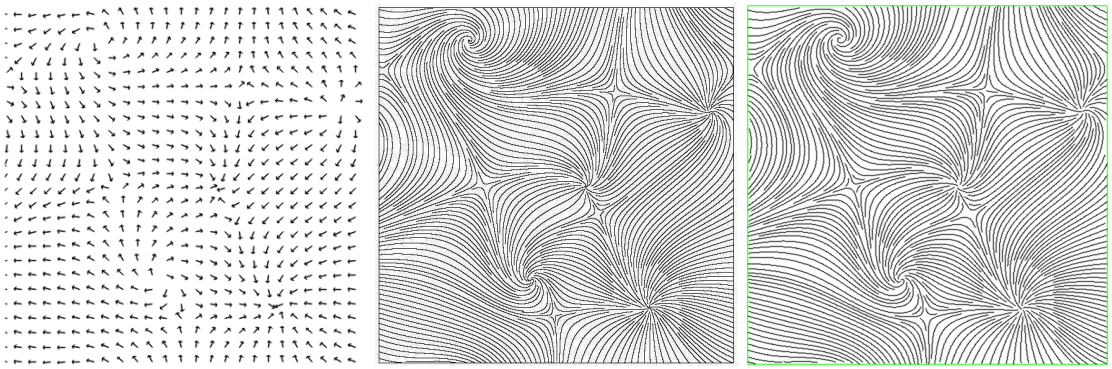
$$\begin{cases} d_{max} = d(\frac{\rho}{2}\,|k_{max}| + (1 - \frac{\rho}{2})\,|k_{min}|) \\ d_{min} = d(\frac{\rho}{2}\,|k_{min}| + (1 - \frac{\rho}{2})\,|k_{max}|) \end{cases}$$

Ce paramètre ρ représente également le degré de confiance que l'on accorde aux observations.

A l'issue de l'intégration des lignes de courbure, certaines régions isotropes restent non comblées. Dans ces zones peu nombreuses, Alliez et al. proposent ainsi d'éffectuer un échantillonnage basé points qui aboutit le plus souvent à des triangulaires. Dans les régions anisotropes, l'orthogonalité des directions principales aboutit naturellement à un maillage largement quadrangulaire. Marinov et Kobbelt [MK04] ont apporté quelques modifications à l'approche de Alliez et al.. Dans leur méthode, l'intégration des lignes ne nécessite plus de paramétrisation car elle se fait directement le long de la surface 3D. Egalement, la construction des deux réseaux de lignes de courbure minimale et maximale se fait de façon simultanée. Les auteurs n'utilisent pas de structure de Delaunay pour les requêtes de proximité mais un système de cache qui leur permet d'interroger à chaque instant les facettes avoisinant le point courant. Enfin, dans les zones isotropes, ils choisissent simplement de prolonger les lignes le long de la direction courante, ce qui leur permet de limiter le nombre de facettes non quadrangulaires.

A.3 Adaptation au rééchantillonnage d'une image

Pour répondre aux objectifs énoncés au début de cette annexe, les travaux précédents nous ont paru intéressants. En effet, dans ces travaux l'espacement entre les lignes de flux dépend de la régularité locale de la surface. Ceci produit des mailles allongées le long



(a) (b) (c)

Fig. A.3 : Résultat de l'algorithme de Mebarki. (a) Un champ de vecteurs initial, (b) le placement de lignes de courant engendré par l'auteur (D'après [Meb04]), (c) le résultat de notre implémentation.

des directions de courbure minimale et contractées le long des directions de courbure maximale. Nous avons tenté d'adapter ces travaux à notre problématique. La première étape a été de développer l'algorithme proposé par Mebarki [Meb04] pour la génération de lignes de flux. La figure A.3 montre un champ vectoriel, les lignes de flux générées par notre implémentation de cet algorithme, et celles générées par l'auteur. Aux différences de paramètres près, nous avons conclu que notre implémentation de cet algorithme était correcte. Nous nous sommes alors attachés à adapter la méthode à notre problème. La première difficulté était de construire un champ de vecteurs reflétant la géométrie d'une image.

A.3.1 Construction du champ vectoriel

La construction du champ vectoriel est une étape cruciale du processus. En effet, ce champ constitue l'attache aux données qui, après intégration, doit nous permettre de générer des lignes adaptées au contenu de l'image. Ce type d'approche est totalement local, l'intégration des lignes dépendant en tout point du champ vectoriel calculé, et se démarque ainsi de l'approche itérative globale que nous avons proposée au chapitre 4. Dans [ACSD+03], Alliez et al. préconisent l'utilisation d'un champ de tenseurs de courbure pour le remaillage de surfaces 3D. Dans notre cas, tout en considérant l'image comme une surface, nous avons cependant privilégié l'extraction d'un champ de gradient plutôt qu'un champ de courbure. En effet, la courbure est une caractéristique intrinsèquement tridimensionnelle qui ne prend pas en compte le point de vue particulier d'une image fixe. Le champ de gradient que nous proposons d'extraire est défini par rapport au plan image. Il comporte 2 attributs :

- La direction de régularité maximale (ou de gradient nul par rapport au plan image) $\mathbf{D_{min}}$, donnée par l'intersection entre le plan image et le plan tangent à la surface 3D. La direction orthogonale est la direction locale de plus fort gradient $\mathbf{D_{max}}$.
- Une mesure de gradient dans la direction $\mathbf{D_{max}}$. La projection d'un cercle de rayon

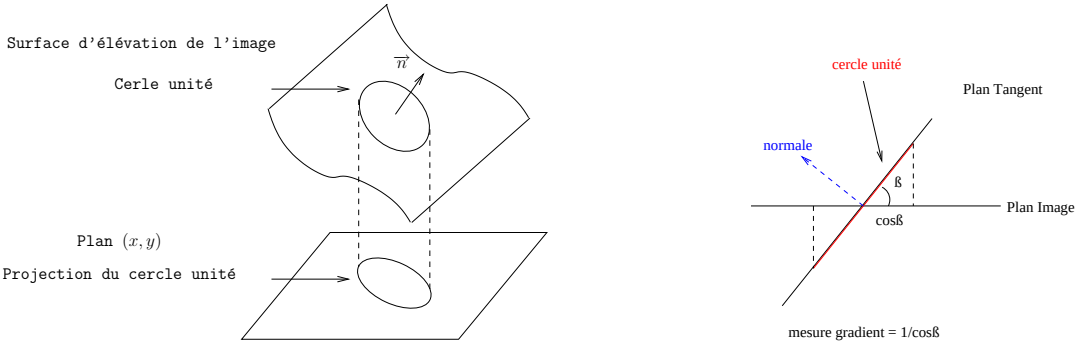


Fig. A.4 : Calcul du champ de gradient. Projection d'un cercle unité du plan tangentiel sur le plan image.

1 du plan tangent sur le plan image nous donne la valeur recherchée, comme illustré sur la figure A.4. Cette projection tient compte du point de vue particulier de l'image. Ainsi, un cercle dessiné sur un plan non parallèle au plan de visualisation sera perçu comme une ellipse. Plus le gradient local est abrupt, plus la longueur du demi-petit axe de l'ellipse est faible.

En calculant ces attributs en chaque pixel de l'image *Lena* **lissée**, nous avons abouti au champ elliptique représenté sur la figure A.5. L'interprétation de ce champ est particulièrement intéressante car il nous donne une idée du noyau d'analyse optimal en chaque point pour le critère choisi (ici, le gradient). Il nous permet ainsi de juger l'erreur qui est commise en appliquant en tout point le noyau d'analyse isotrope des ondelettes classiques. En outre, ce champ elliptique semble bien adapté à la méthode d'Alliez et al. car il est défini directement dans le domaine image (2D).

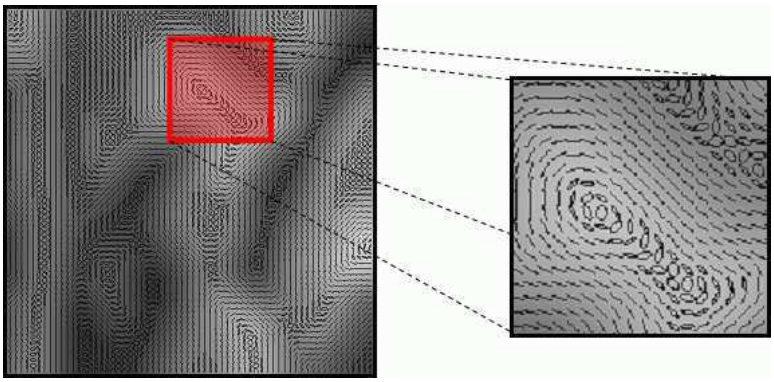


Fig. A.5 : Champ elliptique dense obtenu avec l'image *Lena* lissée.

A partir du champ elliptique obtenu, nous souhaitons intégrer deux réseaux de lignes de flux. Au regard des résultats présentés par Alliez et al., nous avons conclu que le critère déterminant pour aboutir au maillage le plus quadrangulaire et le plus régulier possible est la longueur des lignes. En effet, autour des points nommés « ombilics », où l'intégration des lignes de flux est interrompue, le remaillage quadrangulaire échoue et

la zone est triangularisée. Ainsi, pour optimiser la longueur des lignes, nous souhaitons faire en sorte que l'intégration des lignes ne soit plus interrompue que dans les 2 cas suivants :

1. La ligne croise la frontière du domaine,
2. La ligne s'approche trop d'une ligne existante.

Ceci suppose en particulier qu'une ligne ne puisse pas « boucler » sur elle-même. De ce fait, au lieu de traiter nos deux réseaux de directions principales et secondaires indépendamment comme Alliez et al., nous décidons de décomposer notre champ elliptique en un champ de directions « horizontales » $\mathcal{H}$ et en un champ de directions « verticales » $\mathcal{V}$ tels que :

$$\mathcal{H} = \{\mathbf{D_i} \quad | \quad \mathbf{D_i} \in \mathcal{D}_{max} \cup \mathcal{D}_{min} \quad , \quad |<\mathbf{D_i}, \mathbf{i}>| > |<\mathbf{D_i}, \mathbf{j}>|\},$$

$$\mathcal{V} = \{\mathbf{D_i} \quad | \quad \mathbf{D_i} \in \mathcal{D}_{max} \cup \mathcal{D}_{min} \quad , \quad |<\mathbf{D_i}, \mathbf{i}>| \leqslant |<\mathbf{D_i}, \mathbf{j}>|\},$$

où $\mathbf{D_i}$ est la direction (vecteur normalisé) globalement horizontale ou globalement verticale associée au $i^{ème}$ pixel, $\mathcal{D}_{max}$ et $\mathcal{D}_{min}$ sont les champs de directions de gradient maximal et minimal respectivement, et $\mathbf{i}$ et $\mathbf{j}$ deux vecteurs formant une base orthonormale de l'espace $\mathbb{R}^2$. Sur la figure A.6 sont illustrés les ensembles $\mathcal{D}_{max}$, $\mathcal{D}_{min}$, $\mathcal{H}$ et $\mathcal{V}$ calculés sur l'image Lena lissée. Rappelons que les vecteurs $\mathbf{D_{max_i}}$ et $\mathbf{D_{min_i}}$ correspondent respectivement aux petits axes et grands axes des ellipses introduites précédemment. En particulier nous observons que les vecteurs $\mathbf{D_{min_i}}$ suivent bien les directions de régularité maximale (gradient minimal).

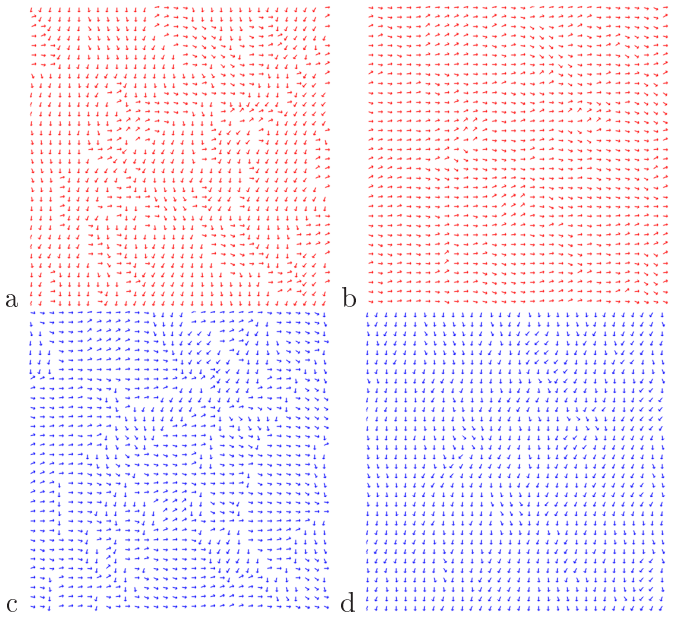


Fig. A.6 : Illustration des ensembles (a) $\mathcal{D}_{max}$, (c) $\mathcal{D}_{min}$, (b) $\mathcal{H}$ et (d) $\mathcal{V}$ calculés sur l'image Lena lissée. $\mathcal{H}$ et $\mathcal{V}$ prennent leurs éléments dans $\mathcal{D}_{max}$ et $\mathcal{D}_{min}$.

Une fois calculés les champs $\mathcal{H}$ et $\mathcal{V}$, nous proposons de lancer séparément sur ces 2 ensembles l'algorithme mis en place par Mebarki, avec toutefois quelques modifications.

A.3.2 Les modifications de l'algorithme

Deux modifications importantes ont été apportées à l'approche de Mebarki [Meb04] : la *gestion des régions isotropes* et les *critères d'arrêts* de l'intégration d'une ligne de courant.

A.3.2.1 Gestion des régions isotropes

Dans l'approche de Mebarki, ainsi que dans l'utilisation qui en est faite par Alliez et al. pour le remaillage de surfaces, une ligne de flux est calculée via une intégration de type Runge-Kutta. A chaque étape de l'intégration, un nouveau vecteur de flux est calculé dans le champ dense pour poursuivre l'intégration. Néanmoins, le champ à traiter est plus complexe dans le cas de Alliez et al. puisque chacun de ces éléments comporte deux directions principales et les deux valeurs de courbures associées. En particulier, ces valeurs de courbures apportent deux contraintes à l'algorithme. D'une part, elles dictent l'espacement local idéal recherché entre les lignes de flux. D'autre part, leur valeur relative dicte le degré d'anisotropie locale. Le cas qui nous intéresse ici est celui où les 2 valeurs de courbure (principale et secondaire) sont très proches. Une région où le champ satisfait cette particularité est une région isotrope où, donc, aucune direction ne prend l'ascendant pour le critère sélectionné (ici, la courbure). Une fois arrivé à l'une de ces régions lors de l'intégration des lignes de courbure maximale, par exemple, il n'est donc pas pertinent de sélectionner une direction en particulier pour poursuivre cette intégration. Alliez et al. décident alors d'interrompre l'intégration (cf. « ombilics »). Ce choix revient à donner aux régions isoptropes un fort poids sémantique pour la surface et d'influencer la structure du maillage en fonction de ces zones, au risque d'aboutir à des mailles non quadrangulaires en plus grand nombre.

Dans notre cas, nous souhaitons absolument privilégier l'obtention d'un maillage quadrangulaire *régulier*. De ce fait, lorsque la ligne en cours d'intégration atteint une zone isotrope, nous choisissons à la manière de Marinov et Kobbelt [MK04] de poursuivre cette intégration en suivant la direction précédente. De façon plus générale, à chaque étape du calcul d'une ligne de courant, nous définissons le vecteur d'intégration courant $\boldsymbol{\gamma}$ relativement au vecteur précédent $\boldsymbol{\gamma_{prev}}$, au vecteur $\mathbf{D_i}$ proposé par le champ de gradient, et à une mesure de confiance au champ. Étant donnée une ellipse de notre champ de gradient, notons l_p la longueur de son demi-petit axe (inversement proportionnelle au gradient local) et l_s la longueur de son demi-grand axe. La mesure de confiance aux données choisie, aussi appelée degré d'isotropie ρ, est alors

$$\rho = l_s/l_p$$

et le vecteur $\boldsymbol{\gamma}$ se calcule à partir de la relation suivante :

$$\boldsymbol{\gamma} = \rho \times \boldsymbol{\gamma_{prev}} + (1 - \rho) \times \mathbf{D_i}. \tag{A.2}$$

Dans le cas où les ellipses sont calculées par projection d'un cercle unitaire du plan tangent sur le plan image, alors $l_p = 1$ et le degré d'isotropie est entièrement défini par l_s : $\rho = l_s$. L'interprétation de l'équation (A.2) est directe :

- Dans une zone isotrope, le paramètre ρ tend vers 1 et l'intégration est entièrement déterminée par la direction précédente $\boldsymbol{\gamma_{prev}}$. Dans la figure A.4(b), ces régions sont celles dont le calcul du gradient a abouti à des cercles.
- Dans une zone fortement anisotrope, le paramètre ρ est proche de 0 et l'intégration dépend alors entièrement de la direction donnée par le champ de gradient. Dans une telle zone, les ellipses sont fortement aplaties, ce qui signifie que le gradient est très prononcé.

En résumé, plus le champ de gradient calculé est anisotrope plus l'intégration des lignes de flux est attaché aux données de ce champ.

A.3.2.2 Critères d'arrêt

Comme nous l'avons indiqué précédemment, la longueur des lignes de flux est déterminante en vue d'obtenir un maillage quadrangulaire régulier. Pour éviter qu'une ligne de flux ne boucle sur elle-même et s'interrompe, nous avons choisi de construire les champs $\mathcal{H}$ et $\mathcal{V}$. Par construction, il est impossible qu'une ligne intégrée via l'un de ces champs ne boucle sur elle-même. Ces champs doivent aboutir à un réseau de lignes « globalement horizontales » et une réseau de lignes « globalement verticales ». Dans l'algorithme décrit par Mebarki, l'intégration d'une ligne de courant est interrompue lorsque cette ligne se rapproche trop d'une ligne déjà existante. De façon à mener à bien les requêtes de proximité, l'auteur préconise l'utilisation d'une structure particulière : une triangulation de Delaunay contrainte (CDT). La CDT est initialisée en plaçant des nœud sur les bords du domaine de visualisation. Lors de l'intégration des lignes de courant, chaque nouveau segment de lignes intégré est inséré comme contrainte dans la triangulation. A une étape donnée, la distance d'un point nouvellement intégré aux autre lignes est la distance de ce point à la contrainte la plus proche. Cette distance est estimée par le diamètre du plus petit cercle parmi tous les cercles circonscrits aux triangles adjacents au point en question.

A.3.3 Résultats-Conclusions

Dans ce paragraphe, nous présentons les résultats et les conclusions auxquels ces recherches ont aboutis. Par souci de clarté, nous récapitulons ci-dessous la démarche suivie :

1. Lissage de l'image,
2. Calcul du champ de gradient par rapport au plan image (figure A.4),
3. Décomposition du champ en deux ensembles $\mathcal{H}$ et $\mathcal{V}$ représentés figure A.6,
4. Intégration via $\mathcal{H}$ d'un réseau de lignes « globalement horizontales »,
5. Intégration via $\mathcal{V}$ d'un réseau de lignes « globalement verticales »,
6. Fusion des 2 réseaux et création du maillage par la méthode décrite dans [ACSD$^+$03].

Les étape 4,5 et 6 appliquées au champ calculé sur l'image *Lena* sont illustrées sur la figure A.7. Nous remarquons d'emblée que la solution proposée n'est pas satisfaisante. Le maillage obtenu en bout de chaîne (figure A.7(d)) est certes à dominance quadrangulaire mais, malgré les modifications faites à l'algorithme pour prolonger les lignes, il demeure fortement irrégulier. Or, transmettre une telle connectivité dans un schéma de codage est prohibitif. Le constat d'une connectivité irrégulière en sortie de l'algorithme n'est

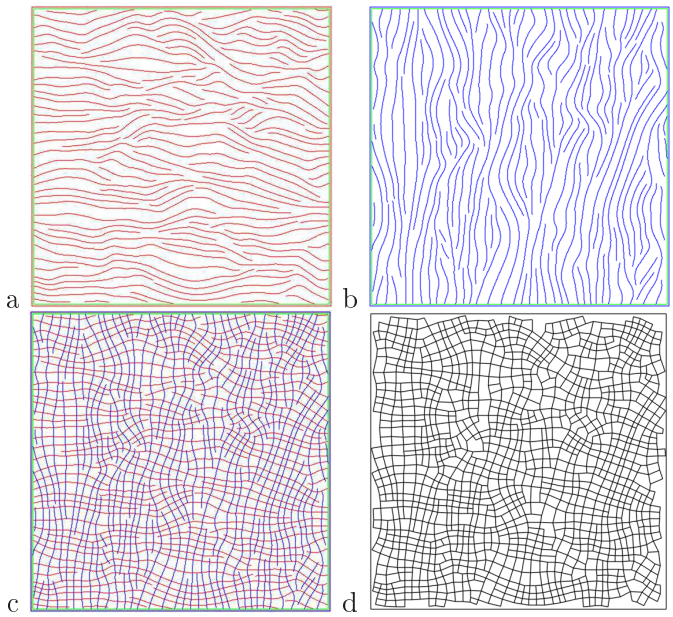


Fig. A.7 : Remaillage de Lena à l'aide des champs $\mathcal{H}$ et $\mathcal{V}$. (a) Lignes de flux issues de $\mathcal{H}$, (b) Lignes de flux issues de $\mathcal{V}$, (c) Fusion des lignes de flux et (d) Maillage final obtenu.

pas le seul élément en défaveur du type d'approche que nous venons d'expérimenter. Ci-dessous, nous commentons les principaux défauts d'une telle approche :

Nécessité d'un lissage préalable. Nous avons mentionné à plusieurs reprises que le calcul du gradient se faisait sur l'image lissée. En effet, la luminance d'une image naturelle est une donnée très bruitée. Calculer le champ sur cette donnée brute conduit à un résultat très chaotique et inexploitable pour l'approche visée. Le lissage de l'image (ou du champ) est donc une étape nécessaire du processus. Dans notre implémentation, nous avons appliqué sur l'image 5 itérations d'un filtre gaussien 7×7. Ce lissage n'est pas sans conséquence puisqu'il efface de nombreux détails de l'image. De manière générale, il semble que cette question du lissage de l'image se posera à chaque fois que nous souhaiterons adapter une approche venant de la 3D. En effet, beaucoup de méthodes utilisées pour traiter des maillages 3D supposent que ces maillages sont la version discrète d'une surface *régulière*. Si cette hypothèse n'est pas vérifiée, le calcul de la courbure n'a par exemple plus de sens.

Complexité de la structure utilisée. Nous avons également mentionné que l'algorithme proposé par Mebarki s'appuie sur une triangulation de Delaunay contrainte pour mener à bien les requêtes de proximité. Dans notre travail, nous avons utilisé la structure fournie par la librairie graphique CGAL (`www.cgal.org`). Même si cette structure est effectivement très efficace, la méthode reste assez lourde. En effet, à chaque fois que l'on souhaite intégrer une nouvelle ligne de courant, le germe initiant cette ligne doit être placé dans la plus grande cavité pour assurer la saturation de l'ensemble du domaine de visualisation. Ceci nécessite la création et la maintenance d'une liste de priorité des facettes de la triangulation de Delaunay. A chaque fois qu'une nouvelle ligne est intégrée, les nouvelles facettes créées en insérant les points de la ligne sont ajoutées à la queue de priorité. L'algorithme s'achève lorsque la queue est vide, signifiant que le domaine est saturé et donc que plus aucune ligne ne doit être ajoutée. Ce procédé reste relativement lourd et l'appliquer à toutes les images d'une vidéo ne semble pas raisonnable.

Manque de robustesse. Ce dernier point est sans doute le plus rédhibitoire. L'objectif initial était d'obtenir un maillage quadrangulaire régulier le plus proche possible des données de l'image. Une des conclusions auxquelles cette étude expérimentale nous a menées est que ce type d'approche *locale* ne semble pas en mesure de répondre à l'objectif de régularité. En effet, avec ce type d'approche, un fort poids est donné à l'attache aux *données* relativement aux contraintes mises en œuvre pour aboutir à un maillage *régulier*. Il n'est donc pas surprenant qu'une telle méthode attachée fortement aux données locales produise une structure irrégulière. Le manque de robustesse vis à vis de l'objectif de régularité condamne donc ce type de méthode : a priori, étant donnée une image naturelle quelconque lissée raisonnablement, il n'est pas possible de prédire le niveau d'irrégularité obtenu en sortie. Un post-traitement pour obtenir la régularité désirée n'est de ce fait pas concevable.

Pour conclure, nous observons que les résultats obtenus avec cette méthode ne répondent pas à nos objectifs. L'intégration de lignes de flux ne permet pas, comme espéré, de refléter la géométrie complexe présente dans une image. Rappelons que notre souhait est de construire un maillage quadrangulaire dont les mailles reflètent *au mieux* la forme du noyau d'analyse à appliquer localement. Ici, les réseaux de lignes de flux ont été intégrés en utilisant uniquement les directions données par les champs. Les valeurs des axes des ellipses, autrement dit les valeurs de gradient, n'ont pas été exploitées, hormis pour le calcul du coefficient de confiance aux données ρ. Or, l'intention initiale était d'utiliser ces valeurs pour construire un réseau de lignes dont la densité dépend du gradient local. Mais la contrainte de densité constante étant déjà difficilement respectée dans le cas « simple », nous n'avons pas essayé d'injecter un critère de densité variable car le résultat n'aurait d'évidence pas été à la mesure de nos attentes.

La méthode présentée dans cette annexe n'a donc pas porté ses fruits. Elle nous a amenés à donner un poids fort à la contrainte de régularité du maillage et à développer l'algorithme d'estimation géométrique décrit au chapitre 4.

Annexe B

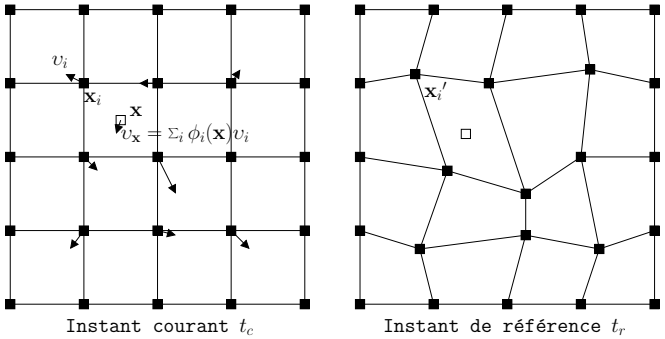
Estimation de mouvement par descente en gradient

Cette annexe présente la technique d'optimisation que nous avons implémentée pour effectuer une estimation de mouvement entre une image à un instant courant t_c et une image à un instant de référence t_r. Le modèle utilisé est le maillage déformable (ou CGI). Les coordonnées $\mathbf{x}_i = (x_i, y_i)$ des nœuds $i \in \{1 \dots N_s\}$ à l'instant courant sont fixées a priori et les paramètres du modèle à estimer sont les déplacements $\Delta\mathbf{x}_i = (\Delta x_i, \Delta y_i)$ de ces nœuds entre l'instant courant et l'instant de référence. Le critère à minimiser est l'erreur entre l'image courante I_{t_c} et sa prédiction $\bar{I}_{t_c} = I_{t_r}(\mathbf{x} + \sum_i \phi_i(\mathbf{x})\Delta\mathbf{x}_i)$. Cette erreur est appelée erreur de « matching » dans [WL94] :

$$\mathbf{E}_m = \sum_{\mathbf{x}} [I_{t_c}(\mathbf{x}) - \bar{I}_{t_c}(\mathbf{x})]^2 \tag{B.1}$$

$$= \sum_{\mathbf{x}} [I_{t_c}(\mathbf{x}) - I_{t_r}(\mathbf{x} + \sum_i \phi_i(\mathbf{x})\upsilon_i)]^2 \tag{B.2}$$

où $\phi_i(\mathbf{x}) = \phi(\mathbf{x} - \mathbf{x}_i)$ est une fonction d'interpolation telle que $\phi_i(\mathbf{x}_i) = 1$. Le support de ϕ définit la région d'influence de chaque nœud.



B.0.3.1 Minimisation de $\mathbf{E}_m$

La descente en gradient permet de minimiser $\mathbf{E}_m$ de façon *globale*. Cette technique est basée sur l'amélioration itérative d'une solution approchée.

Initialement, on a $\{(\Delta x_i, \Delta y_i) = (0,0))\} \quad \forall i \in \{1,..,N_s\}$ et $\bar{I}_{t_c} = \bar{I}_{t_c}^{(0)} = I_{t_r}$.

A chaque itération k, la méthode calcule un déplacement élémentaire des points de contrôle $(\mathrm{d}x_i^{(k)}, \mathrm{d}y_i^{(k)})$ permettant de se rapprocher de la solution optimale.

Plaçons nous à l'issue de l'itération $(k-1)$. Pour tout sommet i, la solution approchée courante est :

$$\begin{pmatrix} \Delta x_i^{(k-1)} \\ \Delta y_i^{(k-1)} \end{pmatrix} = \begin{pmatrix} \sum_{n=1}^{k-1} \mathrm{d}x_i^{(n)} \\ \sum_{n=1}^{k-1} \mathrm{d}y_i^{(n)} \end{pmatrix} \tag{B.3}$$

et l'image compensée correspondante est :

$$\bar{I}_{t_c}^{(k-1)}(x,y) = I_{t_r}(x + \Delta x^{(k-1)}, y + \Delta y^{(k-1)}) \tag{B.4}$$

A l'itération k, on recherche les variations élémentaires $(\mathrm{d}x_i^{(k)}, \mathrm{d}y_i^{(k)})$ qui minimisent l'énergie $\mathbf{E}_m^{(k)}$ donnée par :

$$\mathbf{E}_m^{(k)} = \sum_{(x,y)} [I_{t_c}(x,y) - \bar{I}_{t_c}^{(k-1)}(x + \mathrm{d}x^{(k)}, y + \mathrm{d}y^{(k)})]^2 \tag{B.5}$$

avec

$$\begin{pmatrix} \mathrm{d}x^{(k)} \\ \mathrm{d}y^{(k)} \end{pmatrix} = \begin{pmatrix} \sum_i \phi_i(x,y) \mathrm{d}x_i^{(k)} \\ \sum_i \phi_i(x,y) \mathrm{d}y_i^{(k)} \end{pmatrix} \tag{B.6}$$

Dérivons $\mathbf{E}_m^{(k)}$ par rapport à un paramètre donné, par exemple $\mathrm{d}x_i^{(k)}$, pour un sommet i quelconque. On a :

$$-\frac{1}{2} \frac{\partial \mathbf{E}_m^{(k)}}{\partial \mathrm{d}x_i^{(k)}} = 0$$

$$\Leftrightarrow$$

$$\sum_{(x,y)} \phi_i(x,y) \cdot \frac{\partial \bar{I}_{t_c}^{(k-1)}}{\partial x}(x + \mathrm{d}x^{(k)}, y + \mathrm{d}y^{(k)}) \cdot [I_{t_c}(x,y) - \bar{I}_{t_c}^{(k-1)}(x + \mathrm{d}x^{(k)}, y + \mathrm{d}y^{(k)})] = 0 \tag{B.7}$$

En suivant une approche de type Gauss-Seidel, on cherche à linéariser l'équation (B.7). Considérons le développement limité en (x,y) de $\bar{I}_{t_c}^{(k-1)}(x + \mathrm{d}x^{(k)}, y + \mathrm{d}y^{(k)})$:

$$\begin{aligned}
&\bar{I}_{t_c}^{(k-1)}(x+\mathrm{d}x^{(k)}, y+\mathrm{d}y^{(k)}) \\
&= \\
&\bar{I}_{t_c}^{(k-1)}(x,y) + \sum_j \phi_j(x,y) \cdot \frac{\partial \bar{I}_{t_c}^{(k-1)}}{\partial x}(x,y) \cdot \mathrm{d}x_j^{(k)} + o((\mathrm{d}x_j^{(k)})^2) \\
&\quad + \sum_j \phi_j(x,y) \cdot \frac{\partial \bar{I}_{t_c}^{(k-1)}}{\partial y}(x,y) \cdot \mathrm{d}y_j^{(k)} + o((\mathrm{d}y_j^{(k)})^2)
\end{aligned} \tag{B.8}$$

Si on émet l'hypothèse qu'à chaque itération la descente en gradient génère de petits déplacements ($\ll 1$), alors l'approximation à l'ordre 1 est suffisante.

Le même raisonnement pourrait être mené pour linéariser le terme $\frac{\partial \bar{I}_{t_c}^{(k-1)}}{\partial x}(x + \mathrm{d}x^{(k)}, y + \mathrm{d}y^{(k)})$ dans l'équation (B.7). Néanmoins, on peut raisonnablement négliger l'impact de la dérivée seconde. Dans ce cas [PFTV92], l'approximation à l'ordre 0 suffit, i.e. :

$$\frac{\partial \bar{I}_{t_c}^{(k-1)}}{\partial x}(x+\mathrm{d}x^{(k)}, y+\mathrm{d}y^{(k)}) \approx \frac{\partial \bar{I}_{t_c}^{(k-1)}}{\partial x}(x,y) \tag{B.9}$$

Cette approximation permet en outre de ne pas aboutir à un système quadratique.

En notant $\nabla_x I$ et $\nabla_y I$ les dérivées partielles d'une image en x et y, on montre que l'équation (B.7) peut ainsi être réécrite comme :

$$\begin{aligned}
&\sum_{(x,y)} \sum_j \phi_j(x,y) \cdot \phi_i(x,y) \cdot \nabla_x \bar{I}_{t_c}^{(k-1)}(x,y) \cdot [\nabla_x \bar{I}_{t_c}^{(k-1)}(x,y) \cdot \mathrm{d}x_j^{(k)} + \nabla_y \bar{I}_{t_c}^{(k-1)}(x,y) \cdot y_j^{(k)}] \\
&\qquad = \sum_{(x,y)} \phi_i(x,y) \cdot \nabla_x \bar{I}_{t_c}^{(k-1)}(x,y) \cdot [I_{t_c}(x,y) - \bar{I}_{t_c}^{(k-1)}(x,y)]
\end{aligned} \tag{B.10}$$

L'équation (B.10) correspond à une ligne d'un système linéaire à $2 \times N_s$ inconnues :

$$A \cdot \mathbf{X} = B, \tag{B.11}$$

où $\mathbf{X} = \left\{\mathrm{d}x_1^{(k)}, .., \mathrm{d}x_{N_s}^{(k)}, \mathrm{d}y_1^{(k)}, .., \mathrm{d}y_{N_s}^{(k)}\right\}$.

Les valeurs de la matrice $A = (a_{i,j})$ et des contraintes $B = (b_i)$ sont données par :

$$
\begin{aligned}
\forall (i,j) \in \{1,..,N_s\}^2 & \\
a_{i,j} &= \sum_{(x,y)} \phi_j(x,y) \cdot \phi_i(x,y) \cdot \nabla_x \bar{I}_{t_c}^{(k-1)}(x,y) \cdot \nabla_x \bar{I}_{t_c}^{(k-1)}(x,y) \\
a_{i,j+N_s} &= \sum_{(x,y)} \phi_j(x,y) \cdot \phi_i(x,y) \cdot \nabla_x \bar{I}_{t_c}^{(k-1)}(x,y) \cdot \nabla_y \bar{I}_{t_c}^{(k-1)}(x,y) \\
a_{i+N_s,j} &= \sum_{(x,y)} \phi_j(x,y) \cdot \phi_i(x,y) \cdot \nabla_y \bar{I}_{t_c}^{(k-1)}(x,y) \cdot \nabla_x \bar{I}_{t_c}^{(k-1)}(x,y) \\
a_{i+N_s,j+N_s} &= \sum_{(x,y)} \phi_j(x,y) \cdot \phi_i(x,y) \cdot \nabla_y \bar{I}_{t_c}^{(k-1)}(x,y) \cdot \nabla_y \bar{I}_{t_c}^{(k-1)}(x,y) \\
b_i &= \sum_{(x,y)} \phi_i(x,y) \cdot \nabla_x \bar{I}_{t_c}^{(k-1)}(x,y) \cdot [I_{t_c}(x,y) - \bar{I}_{t_c}^{(k-1)}(x,y)] \\
b_{i+N_s} &= \sum_{(x,y)} \phi_i(x,y) \cdot \nabla_y \bar{I}_{t_c}^{(k-1)}(x,y) \cdot [I_{t_c}(x,y) - \bar{I}_{t_c}^{(k-1)}(x,y)]
\end{aligned}
\tag{B.12}
$$

Lorsque l'intersection des supports respectifs de ϕ_i et ϕ_j est vide, les coefficients $a_{i,j}, a_{i,j+N_s}, a_{i+N_s,j}, a_{i+N_s,j+N_s}$ sont nuls. En général, le support de la fonction de forme ϕ est limité et on aboutit alors à une matrice A creuse à dominante diagonale, symétrique et définie positive. Par exemple si ϕ_i est une fonction bilinéaire qui vaut 1 au nœud i et 0 aux nœuds incidents alors le déplacement d'un nœud n'influence que les mailles incidentes. Chaque ligne de la matrice A comporte uniquement 9 valeurs non nulles. Le système (B.11) peut donc être résolu rapidement avec des techniques itératives (gradient conjugué, méthode de Choleski,...).

B.0.3.2 Augmentation de Levenberg-Marquardt

Certaines valeurs (a_{ii}) sur la diagonale de la matrice A peuvent être très petites ($\ll 1$) lorsque les gradients $\nabla_x \bar{I}_{t_c}$ ou $\nabla_y \bar{I}_{t_c}$ sont très faibles. Dans ce cas, la contrainte b_i est elle aussi très faible et nous aboutissons à une équation du type :

$$\epsilon_1 \cdot x = \epsilon_2, \tag{B.13}$$

avec ϵ_1 et ϵ_2 très faibles. Ce type d'équation rend la résolution du système instable car il peut engendrer de grands déplacements. Or, la linéarisation proposée n'est valable que sous l'hypothèse de petits déplacements.

Pour y remédier, une solution consiste à utiliser une augmentation de Levenberg-Marquardt. Il s'agit de relever les valeurs de la diagonale. Dans nos travaux, nous avons choisi de n'augmenter que les valeurs qui sont en-dessous d'un seuil noté a_{ii}^{norm} :

$$
\begin{aligned}
a_{ii}^{norm} &= (\sum_{(x,y)} \phi_i(x,y) \cdot \phi_i(x,y)) \cdot \nabla_{min}^2 \\
\text{avec} \quad \nabla_{min}^2 &= 1
\end{aligned}
\tag{B.14}
$$

Publications

Conférences internationales

[GPW07a] B. Le Guen, S. Pateux, and J. Weiss. Motion-geometry compensation for analysis-synthesis video coding. Dans *IEEE International Workshop on Multimedia Signal Processing*, pages 300–303, Chania, Crète, Grèce, Octobre 2007.

[GPW07b] B. Le Guen, S. Pateux, and J. Weiss. Non-Geometric Energy Formulation for Adaptive Image Compression. Dans *IEEE International Conference on Image Processing*, pages 161–164, San Antonio, TX, Octobre 2007.

[GPW07c] B. Le Guen, S. Pateux, and J. Weiss. Spatial Anaysis-Synthesis for Improvement of Wavelet Coders. Dans *European Signal Processing Conference*, Poznan, Poland, Septembre 2007.

Conférences nationales

[GPW06] B. Le Guen, S. Pateux, and J. Weiss. Modèle énergétique pour la représentation d'images par ondelettes déformées. Dans *Actes de la conférence CORESA*, pages 30–35, Caen, France, Novembre 2006.

[GPW07] B. Le Guen, S. Pateux, and J. Weiss. Compensation spatio-temporelle globale pour le codage vidéo par ondelettes 3D. Dans *Actes de la conférence CORESA*, Montpellier, France, Novembre 2007.

Brevets

[GP06] B. Le Guen and S. Pateux. Dispositif et procédé de codage et de décodage d'au moins une image, 2006. Brevet FR-06 51 815.

[PGC+07] S. Pateux, B. Le Guen, N. Cammas, I. Amonou, and S. Kervadec. Procédés et dispositifs de codage et de décodage d'une séquence d'images représentée à l'aide de tubes de mouvement, 2007. Brevet FR-756 007.

Bibliographie

[AAAB06] M.A. Agostini, T. André, M. Antonini, and M. Barlaud. Modeling the motion coding error for MCWT video coders. Dans *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, France, Mai 2006.

[ABMD92] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Transactions on Image Processing*, 1(2) :205–220, Avril 1992.

[ACSD+03] P. Alliez, D. Cohen-Steiner, O. Devillers, B. Lévy, and M. Desbrun. Anisotropic polygonal remeshing. *ACM Trans. Graph.*, 22(3) :485–493, 2003.

[ADR96] S. Zahir Azami, P. Duhamel, and O. Rioul. Joint source-channel coding : Panorama of methods. Dans *CNES Workshop on Data Compression*, Toulouse, France, Novembre 1996.

[AhG05] MPEG Wavelet Video AhG. Wavelet codec reference document and software manual. doc. n7334. MPEG document N7334, Poznan MPEG 73th meeting, Juillet 2005.

[AKOK92] C. Auyeung, J. Kosmach, M. Orchard, and T. Kalafatis. Overlapped block motion compensation. Dans *SPIE Visual Communication on Image Processing*, volume 1818, pages 561–572, Novembre 1992.

[Alp92] B.K. Alpert. *Wavelets and other bases for fast numerical linear algebra.* C.K. Chui, editor, Academic Press, New York, 1992.

[Alt97] Y. Altunbasak. Object-scalable mesh-based coding of synthetic and natural image objects. Dans *IEEE International Conference on Image Processing*, volume 3, pages 94–97, Santa Barbara, CA, Octobre 1997.

[And07] T. André. *Codage vidéo scalable par transformée en ondelettes et mesure de distortion entropique.* Thèse de Doctorat, Université de Nice-Sophia Antipolis, Septembre 2007.

[AR02] M. Antonini and V. Ricordel. *Traité IC2*, chapitre Quantification, pages 45–72. Hermès, Paris, Janvier 2002.

[AS98] P.K. Agarwal and S. Suri. Surface Approximation and Geometric Partitions. *SIAM Journal on Computing*, 27(4) :1016–1035, Août 1998.

[ASH87] E.H. Adelson, E. Simoncelli, and R. Hingorani. Orthogonal pyramid transforms for image coding. Dans *SPIE Visual Communication on Image Processing*, volume 845, pages 50–58, Cambridge, MA, Octobre 1987.

[AT97a] Y. Altunbasak and A.M. Tekalp. Closed-form connectivity-preserving solutions for motion compensation using 2D meshes. *IEEE Transactions on Image Processing*, 6(9) :1255–1269, Septembre 1997.

[BA83] P.J. Burt and E.H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4) :532–540, Avril 1983.

[Bal05] R. Balter. *Construction d'un maillage 3D évolutif et scalable pour la vidéo.* Thèse de Doctorat, Université de Rennes 1, Mai 2005.

[BGM06] Raphaèle Balter, Patrick Gioia, and Luce Morin. Scalable and efficient coding using 3D modeling. *IEEE Transactions on Multimedia*, 8 :1147–1155, Décembre 2006.

[BM03] R. Balter and L. Morin. Morphing 3D bidirectionnel entièrement automatique. Dans *Actes de la conférence ORASIS*, pages 363–371, Gérardmer, France, may 2003.

[Bra05b] S. Brangoulo. *Codage d'images fixes et vidéos par ondelettes de seconde génération. Théorie et applications.* Thèse de Doctorat, Université de Rennes 1, 2005.

[BT97] P.J.L. Van Beek and A.M. Tekalp. Object-based video coding using forward tracking 2D mesh layers. Dans *SPIE Visual Communication on Image Processing*, volume 3024, pages 699–710, San Jose, CA, Février 1997.

[BT.02] BT.500-11. Methodology for the subjective assessment of the quality of television picture. ITU-R Recommendation, 2002.

[BTU01] T. Blu, P. Thevenaz, and M. Unser. MOMS : Maximal-order interpolation of minimal support. *IEEE Transactions on Image Processing*, 10(7) :1069–1080, 2001.

[Cam04b] N. Cammas. *Codage vidéo scalable par maillages et ondelettes t+2D.* Thèse de Doctorat, Université de Rennes 1, 2004.

[Can98] E. J. Candès. *Ridgelets : theory and applications.* Thèse de Doctorat, Department of Statistics, Stanford University, 1998.

[CCB03] M. Carnec, P. Le Callet, and D. Barba. Full reference and reduced reference metrics for image quality assessment. Dans *IEEE International Symposium on Signal Processing and Its Applications*, volume 1, pages 477–480, Paris, France, Juillet 2003.

[CD99a] E. J. Candès and D. L. Donoho. *Curvelets - A Surprisingly Effective Nonadaptive Representation for Objects with Edges.* Vanderbilt University Press, Nashville, TN, 1999.

[CD99b] E. J. Candès and D. L. Donoho. Ridgelets : a key to higher-dimensional intermittency ? *Roy Soc of London Phil Tr A*, 357(1760) :2495–2509, Septembre 1999.

[CDF89a] A. Cohen, I. Daubechies, and J.C. Feauveau. Biorthogonal bases of compactly supported Wavelets. Rapport technique TM 11217-900529-07, AT&T Bell Laboratories, 1989.

[CG05] V. Chappelier and C. Guillemot. Oriented wavelet transform on a quincunx pyramid for image compression. Dans *IEEE International Conference on Image Processing*, Septembre 2005.

[CGM04b] V. Chappelier, C. Guillemot, and S. Marinkovic. Image coding with iterated contourlet and wavelet transforms. Dans *IEEE International Conference on Image Processing*, Octobre 2004.

[Cha05b] V. Chappelier. *Codage progressif d'images par ondelettes orientées*. Thèse de Doctorat, Université de Rennes 1, 2005.

[CHRW03] P. Chen, K. Hanke, T. Rusert, and J.W. Woods. Improvements to the MC-EZBC scalable video coder. Dans *IEEE International Conference on Image Processing*, volume 2, pages 81–84, Barcelona, Spain, Septembre 2003.

[CP03b] N. Cammas and S. Pateux. Fine grain scalable video coding using 3D wavelets and active meshes. Dans *SPIE Visual Communication on Image Processing*, Santa Clara, CA, Janvier 2003.

[CVGPC06] P. Le Callet, C. Viard-Gaudin, S. Péchard, and E. Caillault. No reference and reduced reference video quality metrics for end to end QoS monitoring. *IEICE Transactions on Communications*, E89-B(2) :289–296, Février 2006.

[CW96] M.C. Chen and A.N. Willson. Motion vector optimization of control grid interpolation and overlapped block motion compensation using iterative dynamic programming. Dans *European Signal Processing Conference*, volume 2, pages 1095–1098, Trieste, Italy, Septembre 1996.

[CW99] S-J. Choi and J.W. Woods. Motion-compensated 3D subband coding of video. *IEEE Transactions on Image Processing*, 8(2) :155–167, Février 1999.

[Dau88] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41 :909–996, 1988.

[Dau92] I. Daubechies. *Ten lectures on wavelets*, volume 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM Ed., Philadelphia, 1992.

[DD95] D.F. Dementhon and L.S. Davis. Model-based object pose in 25 lines of code. *ACM International Journal of Computer Vision*, 15(1-2) :123–141, 1995.

[DDI06] L. Demaret, N. Dyn, and A. Iske. Image compression by linear splines over adaptive triangulations. *Signal Processing*, 86(7) :1604–1616, Juillet 2006.

[Dee95] M. Deering. Geometry compression. Dans *Computer Graphics*, volume 29 of *Annual Conference Series*, pages 13–20, 1995.

[DeV98] R. DeVore. Nonlinear Approximation. *Acta Numerica 7*, pages 51–150, 1998.

[DLG90] N. Dyn, D. Levin, and J. Gregory. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions on Graphics*, 9(2) :160–169, Avril 1990.

[Do01b] M. N. Do. *Directional Multiresolution Image Representation.* Thèse de Doctorat, Department of Communication Systems, Swiss Federal Institute of Technology Lausanne, Novembre 2001.

[Don99] D.L. Donoho. Wedgelets : Nearly Minimax Estimation of Edges. *The Annals of Statistics*, 27(3) :859–897, Juin 1999.

[Don00] D.L. Donoho. Orthonormal ridgelets and linear singularities. *SIAM Journal on Mathematical Analysis*, 31(5) :1062–1099, Avril 2000.

[DS98] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.*, 4(3) :247–269, 1998.

[DV03b] M.N. Do and M. Vetterli. The finite ridgelet transform for image representation. *IEEE Transactions on Image Processing*, 12 :16–28, Janvier 2003.

[DV05] M. N. Do and M. Vetterli. The Contourlet Transform : An Efficient Directional Multiresolution Image Representation. *IEEE Transactions on Image Processing*, 14(12) :2091–2106, Décembre 2005.

[DWL04] W. Ding, F. Wu, and S. Li. Lifting-based Wavelet Transform with Directionally Spatial Prediction. Dans *Proc. Picture Coding Symposium*, San Francisco, CA, Décembre 2004.

[DWW+07] W. Ding, F. Wu, X. Wu, S. Li, and H. Li. Adaptive Directional Lifting-Based Wavelet Transform for Image Coding. *IEEE Transactions on Image Processing*, 16(2) :416–427, Février 2007.

[EB98] M. P. Eckert and A. P. Bradley. Perceptual quality metrics applied to still image compression. *Signal Processing*, 70 :177–200, Novembre 1998.

[EF01] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. Dans *ACM SIGGRAPH*, pages 341–346, New York, NY, 2001. ACM Press.

[ER04] R. Eslami and H. Radha. Wavelet-based Contourlet Coding Using an SPIHT-like algorithm. Dans *IEEE International Conference on Image Processing*, Octobre 2004.

[FH05] M.S. Floater and K. Hormann. Surface parameterization : a tutorial and survey. Dans N. A. Dodgson, M. S. Floater, and M. A. Sabin, editors, *Advances in multiresolution for geometric modelling*, pages 157–186. Springer Verlag, 2005.

[FHH93] D. J. Field, A. Hayes, and R. F. Hess. Contour integration by the human visual system : evidence for a local "association field". *Vision Research*, 33(2) :173–193, Janvier 1993.

[Fie87] D. J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America*, 4 :2379–2394, 1987.

[Fie93] D. J. Field. *Scale-invariance and self-similar 'wavelet' transforms : An analysis of natural scenes and mammalian visual systems.*, chapitre Wavelets, fractals and Fourier transforms, pages 151–193. Clarendon Press, Oxford, 1993.

[FK05] R. Ferzli and L.J. Karam. No-reference objective wavelet based noise immune image sharpness metric. Dans *IEEE International Conference on Image Processing*, volume 1, pages 405–408, Genova, Italy, Septembre 2005.

[FL96] H. Le Floch and C. Labit. Irregular image sub-sampling and reconstruction by adaptive sampling. Dans *IEEE International Conference on Image Processing*, volume 3, pages 379–382, Lausanne, Switzerland, Septembre 1996.

[Gal02] F. Galpin. *Représentation 3D de séquences vidéo : Schéma d'extraction automatique d'un flux de modèles 3D, applications à la compression et à la réalité virtuelle.* Thèse de Doctorat, Université de Rennes 1, Janvier 2002.

[GBMA04] B. Le Guen, R. Balter, L. Morin, and P. Alliez. Morphing de modèles 3D estimés. Dans *Actes de la conférence CORESA*, Lille, Mai 2004.

[GFS97] B. Girod, N. Färber, and E. Steinbach. Performance of the H.263 video compression standard. *Journal of VLSI Signal Processing : Systems for Signal, Image, and Video Technology. Special Issue on Recent Development in Video : Algorithms, Implementation and Applications*, 17(2/3) :101–111, Novembre 1997.

[GH97] M. Garland and P. Heckbert. Surface simplification using quadric error metrics. Dans *SIGGRAPH*, volume 31, pages 99–108, Los Angeles, CA, Août 1997.

[Gha90] M. Ghanbari. The cross-search algorithm for motion estimation. *IEEE Transactions on Communications*, 38(7) :950–953, Juillet 1990.

[Gir93] B. Girod. What's wrong with mean-squared error ? Dans Massachussetts : The MIT Press A. B. Watson, Ed. Cambridge, editor, *Digital Images and Human Vision*, pages 207–220, 1993.

[GM02] F. Galpin and L. Morin. Sliding adjustment for 3D video representation. *EURASIP Journal on Applied Signal Processing*, 2002(1) :1088–1101, Janvier 2002.

[Gra84] R. M. Gray. Vector quantization. *IEEE ASSP Magazine*, pages 4–29, Avril 1984.

[GVSS00] I. Guskov, K. Vidimce, W. Sweldens, and P. Schröder. Normal meshes. Dans Kurt Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 95–102. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[H.203] ITU-T Recommendation H.264. Advanced video coding for generic audio-visual services, Mai 2003.

[HB95] D.J. Heeger and J.R. Bergen. Pyramid-based texture analysis/synthesis. Dans *ACM SIGGRAPH*, pages 229–238, 1995.

[HDD+93] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. Dans *SIGGRAPH*, volume 27, pages 19–26, Anaheim, CA, Août 1993.

[HH94] C.L. Huang and C.Y. Hsu. A new motion compensation method for image sequence coding using hierarchical grid interpolation. *IEEE Transaction on Circuits and Systems for Video Technology*, 4(1) :42–52, Février 1994.

[HMCP01] G. Heising, D. Marpe, H. L. Cycon, and A. P. Petukhov. Wavelet-based very low bit-rate video coding using image warping and overlapped block motion compensation. *IEE Proceedings - Vision, Image, and Signal Processing*, 148(2) :93–101, Avril 2001.

[Hop96] H. Hoppe. Progressive Meshes. Dans *SIGGRAPH*, volume 30, pages 99–108, New Orleans, Louisiana, Août 1996.

[Huf52] D.A. Huffman. A method for the construction of minimum-redundancy codes. Dans *Proceedings of the Institute of Radio Engineers*, volume 40, pages 1098–1101, Massachusetts Institute of Technology, Cambridge, Mass., Septembre 1952.

[HW01a] S.T. Hsiang and J.W. Woods. Embedded image coding using zeroblocks of subband/wavelet coefficients and context modeling. Dans *IEEE Data Compression Conference*, pages 83–92, 2001.

[HW01b] S.T. Hsiang and J.W. Woods. Embedded video coding using invertible motion compensated 3D subband/wavelet filter bank. *Signal Processing : Image Communication*, 16 :705–724, Mai 2001.

[IM00] P. Ishwar and P. Moulin. On spatial adaptation of motion-field smoothness in video coding. *IEEE Transaction on Circuits and Systems for Video Technology*, 10(6) :980–989, Septembre 2000.

[JCLB01] M. Jansen, H. Choi, S. Lavu, and R. Baraniuk. Multiscale Image Processing Using Normal Triangulated Meshes. Dans *IEEE International Conference on Image Processing*, Thessaloniki, Greece, Octobre 2001.

[JJ81] I.R. Jain and A.K. Jain. Displacement measurement and its application in interframe image coding. *IEEE Transactions on Communications*, COM-29 :1799–1808, Décembre 1981.

[JL97] B. Jobard and W. Lefer. Creating evenly-spaced streamlines of arbitrary density. Dans W. Lefer and M. Grave, editors, *Visualization in Scientific Computing '97. Proceedings of the Eurographics Workshop in Boulogne-sur-Mer, France*, pages 43–56, Wien, New York, 1997. Springer Verlag.

[JRB07a] G. Jeannic, V. Ricordel, and D. Barba. The edge driven oriented wavelet transform : An anisotropic multidirectional representation with oriented

lifting scheme. Dans *SPIE Visual Communication on Image Processing*, San Jose, CA, USA, Janvier 2007.

[JRB07b] G. Jeannic, V. Ricordel, and D. Barba. A multiresolution approach for the coding of edges of still images using adaptive arithmetic coding. Dans *Picture Coding Symposium*, Lisbonne, Portugal, Novembre 2007.

[JRB07c] G. Jeannic, V. Ricordel, and D. Barba. Représentation structurelle d'images fixes par transformée en ondelettes orientées. Dans *Actes du Colloque GRETSI sur le Traitement du Signal et des Images*, Troyes, France, Septembre 2007.

[JVT06] JVT - ISO/IEC 14496-10 AVC - ITU-T Recommendation H.264 Amendment 3. *Advanced Video Coding Amendment 3 : Scalable Video Coding.* Final Draft International Standard of H.264/AVC Scalable Video Coding Amendment, juillet 2006.

[KCK96] T.-Y. Kuo, J. Chalidabhongse, and C.-C. Kuo. Fast motion vector search for overlapped block motion compensation. Dans *Thirtieth Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 948–952, Pacific Grove, CA, Novembre 1996.

[KIH+81] T. Koga, K. Ilinuma, A. Hirano, Y. Iijima, and T. Ishiguro. Motion compensated interframe coding for video conferencing. Dans *National Telesystems Conference*, volume C, pages 961–965, New Orleans, LA, Décembre 1981.

[Kin98] N.G. Kingsbury. The dual-tree complex wavelet transform : a new efficient tool for image restoration and enhancement. Dans *European Signal Processing Conference*, pages 319–322, Island of Rhodes, Greece, Septembre 1998.

[KXP00] B.J. Kim, Z. Xiong, and W.A. Pearlman. Low bit-rate, scalable video coding with 3D set partitioning in hierarchical trees (3D SPIHT). *IEEE Transaction on Circuits and Systems for Video Technology*, 10(8) :1374–1387, Décembre 2000.

[LCL+02] M.-C. Lee, W.-G. Chen, C.-L. Lin, C. Gu, T. Markok, S.I. Zabinsky, and R. Szeliski. A layered video object coding system using sprite and affine motion model. *IEEE Transaction on Circuits and Systems for Video Technology*, 7(1) :130–145, Février 2002.

[LDW97] M. Lounsbery, T.D. DeRose, and J. Warren. Multiresolution Analysis for Surfaces of Arbitrary Topological Type. *ACM Transactions on Graphics*, 16(1) :34–73, Janvier 1997.

[Lec99b] P. Lechat. *Représentation et codage de séquences vidéo par maillages 2D déformables.* Thèse de Doctorat, Université de Rennes 1, Octobre 1999.

[LF96] L.K. Liu and E. Feig. A block-based gradient descent search algorithm for block motion estimation in video coding. *IEEE Transaction on Circuits and Systems for Video Technology*, 6(4) :419–422, Août 1996.

[LLL+01] L. Luo, J. Li, S. Li, Z. Zhuang, and Y.-Q. Zhang. Motion compensated lifting wavelet and its application in video coding. Dans *IEEE International Conference on Multimedia and Expo*, pages 365–368, Tokyo, Japan, Août 2001.

[LLMD06] C. Laurent, N. Laurent, M. Maurizot, and T. Dorval. In depth analysis and evaluation of saliency-based color image indexing methods using wavelet salient features. *Multimedia Tools Applications*, 31(1) :73–94, 2006.

[LLo82] S.P. LLoyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28 :129–137, Mars 1982.

[LLS99] P. Lechat, N. Laurent, and H. Sanson. Scalable image coding with fine granularity based on hierarchical mesh. Dans *SPIE Visual Communication on Image Processing*, volume 3653, pages 1130–1142, San José, CA, Janvier 1999.

[LM01] J. Liu and P. Moulin. Information-theoretic analysis of interscale and intrascale dependencies between image wavelet coefficients. *IEEE Transactions on Image Processing*, 10(10) :1647–1658, Novembre 2001.

[Loo87a] C.T. Loop. Smooth subdivision surfaces based on triangles. Master's thesis, Departement of Mathematics, University of Utah, Août 1987.

[Loo87b] C.T. Loop. Smooth subdivision surfaces based on triangles. Rapport de DEA, Departement of Mathematics, University of Utah, Août 1987.

[LT98] P. Lindstrom and G. Turk. Fast and memory efficient polygonal simplification. Dans *IEEE Visualization*, pages 279–286, 1998.

[LW95] O. Lee and Y. Wang. Non-uniform image sampling and interpolation over deformed meshes and its hierarchical extension. Dans *SPIE Visual Communication on Image Processing*, pages 389–400, Taipei, Taiwan, Mai 1995.

[LWLZ03] L. Luo, F. Wu, S. Li, and Z. Zhuang. Advanced Lifting-Based Motion-Threading Technique for the 3D Wavelet Video Coding. Dans *SPIE Visual Communication on Image Processing*, Juillet 2003.

[LZL94] R. Li, B. Zeng, and M. Liou. A new three-step search algorithm for block motion estimation. *IEEE Transaction on Circuits and Systems for Video Technology*, 4(4) :438–442, Août 1994.

[MAD05] A. Mebarki, P. Alliez, and O. Devillers. Farthest point seeding for efficient placement of streamlines. Dans *IEEE Conference on Visualization*, pages 479–486, Minneapolis, MN, Octobre 2005.

[Mal89] S. Mallat. A theory of multiresolution signal decomposition : the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11 :674–693, Juillet 1989.

[Mal99] S. Mallat. *A Wavelet Tour of Signal Processing.* Academic Press, 1999.

[Mar82] D. Marr. *Vision : A Computational Investigation into the Human Representation and Processing of Visual Information.* W.H. Freeman, San Francisco, 1982.

[Mar00] G. Marquant. *Représentation par maillage adaptatif déformable pour la manipulation et la communication d'objets vidéo.* Thèse de Doctorat, Université de Rennes 1, Décembre 2000.

[MDF+99] J. Mendola, A. Dale, B. Fischl, A. Liu, and R. Tootell. The representation of illusory and real contours in human cortical visual areas revealed by functional MRI. *Journal of Neuroscience*, 19(19) :8560–8572, Octobre 1999.

[Meb04] A. Mebarki. Placement de lignes de courant. Rapport de DEA, Université de Nice Sophia-Antipolis, Septembre 2004.

[Mey88] Y. Meyer. Construction de bases orthonormées d'ondelettes. *Revista Matemática Iberoamericana*, 4(1) :31–39, 1988.

[MF98] S. Mallat and F. Falzon. Analysis of low bit rate image transform coding. *IEEE Transactions on Image Processing*, 46(4), Avril 1998.

[MK04] M. Marinov and L. Kobbelt. Direct anisotropic quad-dominant remeshing. Dans *Pacific Conference on Computer Graphics and Applications*, pages 207–216, 2004.

[MPE93] MPEG-1. Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s. Dans *Doc. ISO/IEC 11172-2 Video*, 1993.

[MPE94] MPEG-2. Generic coding of moving pictures and associated audio information. Dans *Doc. ISO/IEC 13818-2 Video*, 1994.

[MPE02] MPEG-4. MPEG-4 overview, coding of moving pictures and audio. Doc. ISO/IEC JTC1/SC29/WG11, Mars 2002.

[MPL00b] G. Marquant, S. Pateux, and C. Labit. Mesh-based scalable image coding with rate-distortion optimization. Dans *SPIE Image and Video Communications and Processing*, pages 101–110, San Jose, CA, Avril 2000.

[MPL00c] G. Marquant, S. Pateux, and C. Labit. Multi-resolution Mesh-Based motion estimation using a "backward in forward" tracking method". Dans *European Signal Processing Conference*, Tampere, Finland, Septembre 2000.

[MT06] N. Mehrseresht and D. Taubman. A flexible structure for fully scalable motion-compensated 3D DWT with emphasis on the impact of spatial scalability. *IEEE Transactions on Image Processing*, 15(3) :740–753, Mars 2006.

[MZ93] S. Mallat and Z. Zhang. Matching pursuit with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41 :3397–3415, Décembre 1993.

[NH91] Y. Nakaya and H. Harashima. An iterative motion estimation method using triangular patches for motion compensation. Dans *SPIE Visual Communication on Image Processing*, volume 1605, pages 546–557, Boston, MA, Novembre 1991.

[NO92] S. Nogaki and M. Ohta. An overlapped block motion compensation for high quality motion picture coding. Dans *IEEE International Symposium on Circuits and Systems*, volume 1, pages 184–187, 1992.

[OF96] B.A. Olshausen and D.J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381 :607–609, 1996.

[Ohm94] J.R Ohm. Three-dimensional subband coding with motion compensation. *IEEE Transactions on Image Processing*, 3(5) :559–571, Septembre 1994.

[OS94] M.T. Orchard and G.J. Sullivan. Overlapped block motion compensation : an estimation-theoretic approach. *IEEE Transactions on Image Processing*, 3(5) :693–699, Septembre 1994.

[Pau06] G. Pau. *Ondelettes et décompositions spatio-temporelles avancées ; application au codage vidéo scalable.* Thèse de Doctorat, ENST Paris, Juin 2006.

[Pen02] E. Le Pennec. *Bandelettes et représentation géométrique des images.* Thèse de Doctorat, Ecole Polytechnique, Décembre 2002.

[Pey05b] G. Peyré. *Géométrie multi-échelles pour les images et les textures.* Thèse de Doctorat, Ecole Polytechnique, Décembre 2005.

[PFTV92] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C, The Art of Scientific Computing.* Cambridge University Press, 1992.

[PM00] E. Le Pennec and S. Mallat. Image Compression with Geometrical Wavelets. Dans *IEEE International Conference on Image Processing*, volume 1, pages 661–664, Vancouver, BC, Canada, Septembre 2000.

[PM03] E. Le Pennec and S. Mallat. Non linear image approximation with bandelets. Rapport technique, CMAP/École Polytechnique, 2003.

[PM05] E. Le Pennec and S. Mallat. Sparse Geometric Image Representations with Bandelets. *IEEE Transactions on Image Processing*, 14(4) :423–438, Avril 2005.

[PMCM01] S. Pateux, G. Marquant, and D. Chavira-Martinez. Object mosaicking via meshes and cracklines technique. Application to low bit-rate video coding. Dans *Picture Coding Symposium*, Seoul, Korea, Avril 2001.

[PPB01] B. Pesquet-Popescu and V. Bottreau. Three-dimensional lifting schemes for motion compensated video compression. Dans *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1793–1796, Salt Lake City, UT, Mai 2001.

[PPPP05] G. Piella, G. Pau, and B. Pesquet-Popescu. Adaptive lifting schemes combining seminorms for lossless image compression. Dans *IEEE International Conference on Image Processing*, pages 753–756, Genève, Septembre 2005.

[PS00a] T. Pappas and R. Safranek. Perceptual criteria for image quality evaluation. Dans Al Bovik, editor, *Handbook of Image and Video Processing*, pages 669–684. Academic Press, San Diego, 2000.

[Rad17] J. Radon. Über die bestimmung von funktionen durch ihre integralwerte längs gewisser mannigfaltigkeiten. *Sächsische Akademie der Wissenschaften, Leipzig. Math. Nat.*, 69 :262–277, 1917.

[Ram93b] K. Ramchandran. *Joint Optimization Techniques in Image and Video Coding with Applications to Multiresolution Digital Broadcast.* Thèse de Doctorat, Columbia University, New York, NY, 1993.

[RAPP06] A. Robert, I. Amonou, and B. Pesquet-Popescu. Amélioration de codeurs DCT par orientations des blocs de la transformée. Dans *Actes de la conférence CORESA*, Caen, France, Novembre 2006.

[RAPP07] A. Robert, I. Amonou, and B. Pesquet-Popescu. Amélioration du codage H.264 par orientation des blocs de la transformée. Dans *Actes de la conférence CORESA*, Montpellier, France, Novembre 2007.

[RG98] D.L. Neuhoff R.M. Gray. Quantization. *IEEE Transactions on Information Theory*, 44 :2325–2384, Octobre 1998. Shannon Commemorative Issue, 1948-1998.

[Ric03] I.G. Richardson. Prediction of inter macroblocks in P-slices. H.264/MPEG-4 Part 10 White Paper, 2003. http ://www.vcodex.com/files/h264_interpred.pdf.

[Rob08] A. Robert. *Schéma de codage vidéo hybride.* Thèse de Doctorat, Ecole Nationale Supérieure des Télécommunications, Février 2008. A paraître.

[RWB02] J.K. Romberg, M.B. Wakin, and R.G. Baraniuk. Multiscale Wedgelet Image Analysis : Fast Decompositions and Modeling. Dans *IEEE International Conference on Image Processing*, volume 3, pages 585–588, Rochester, New York, Juin 2002.

[Sai03] A. Said. Arithmetic coding. Dans Ed. K. Sayood, editor, *Lossless Compression Handbook*, San Diego, CA, 2003. Academic Press.

[SB91] G.J. Sullivan and R.L. Baker. Motion compensation for video compression using control grid interpolation. Dans *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 2713–2716, Toronto, Ont., Canada, Avril 1991.

[SCD02] J. Starck, E. J. Candés, and D. L. Donoho. The Curvelet transform for image denoising. *IEEE Transactions on Image Processing*, 11(6) :670–684, Juin 2002.

[SCE01] E. Skodras, C. Christopoulos, and T. Ebrahimi. The JPEG 2000 still image compression standard. *IEEE Signal Processing Magazine*, 18 :36–58, Septembre 2001.

[SG193] ITU-T SG15. Video codec for audiovisual service at Px64 Kbits/s. Dans *ITU-T recommendation H.261 Version 3*, Mars 1993.

[Sha48] C. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27 :379–423, 623–656, 1948.

[Sha93] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41(12) :3445–3462, Décembre 1993.

[Sik97] T. Sikora. MPEG digital video-coding standards. *IEEE Signal Processing Magazine*, 14(5) :82–100, Septembre 1997.

[SM97] J.K. Su and R.M. Mersereau. Non-iterative rate-constrained motion estimation for OBMC. Dans *IEEE International Conference on Image Processing*, pages 33–36, Santa Barbara, CA, Octobre 1997.

[SM00] J.K. Su and R.M. Mersereau. Motion estimation methods for overlapped block motion compensation. *IEEE Transactions on Image Processing*, 9(9) :1509–1521, Septembre 2000.

[SMW07] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Transaction on Circuits and Systems for Video Technology*, 17(9) :1103–1120, September 2007.

[SP96] A. Said and W.A. Pearlman. A new fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Transaction on Circuits and Systems for Video Technology*, 6 :243–250, Juin 1996.

[SS96] W. Sweldens and P. Schröder. Building your own wavelets at home. *Wavelets in Computer Graphics*, 1996. ACM SIGGRAPH Course Notes.

[ST01] A. Secker and D. Taubman. Motion-compensated highly scalable video compression using an adaptive 3D wavelet transform based on lifting. Dans *IEEE International Conference on Image Processing*, volume 2, pages 1029–1032, Octobre 2001.

[ST04] A. Secker and D. Taubman. Highly scalable video compression with scalable motion coding. *IEEE Transactions on Image Processing*, 13(8) :1029–1041, Août 2004.

[Swe97] W. Sweldens. The lifting scheme : A construction of second generation wavelets. *SIAM Journal on Mathematical Analysis*, 29(2) :511–546, 1997.

[Tau99] D. Taubman. High performance scalable image compression with EBCOT. *IEEE Transactions on Image Processing*, 9(7) :1158–1170, Juillet 1999.

[Tek95] M. Tekalp. *Digital video processing.* Prentice Hall Signal Processing Series, 1995.

[TG98] C. Touma and C. Gotsman. Triangle mesh compression. Dans *Graphics Interface*, pages 26–34, San Francisco, CA, Juin 1998.

[TM01] D. Taubman and M. Marcellin. *JPEG2000 : Image compression fundamentals, standards and pratice.* Kluwer Academic Publishers, Novembre 2001.

[TR98] G. Taubin and J. Rossignac. Geometric compression through topological surgery. *ACM Transactions on Graphics*, 17(2) :84–115, Avril 1998.

[TV91] D. Terzopoulos and M. Vasilescu. Sampling and Reconstruction with Adaptive Meshes. Dans *Proc. IEEE Computer Vision and Pattern Recognition Conference*, pages 70–75, Lahaina, HI, 1991.

[TZ94a] D. Taubman and A. Zakhor. Multirate 3D subband coding of video. *IEEE Transactions on Image Processing*, 3(5) :572–588, Septembre 1994.

[TZ94b] D. Taubman and A. Zakhor. Orientation adaptive subband coding of images. *IEEE Transactions on Image Processing*, 3 :421–436, Juillet 1994.

[VBLVD06] V. Velisavljevic, B. Beferull-Lozano, M. Vetterli, and P.L. Dragotti. Directionlets : anisotropic multi-directional representation with separable filtering. *IEEE Transactions on Image Processing*, Juillet 2006.

[Vel05b] V. Velisavljevic. *Directionlets : anisotropic multi-directional representation with separable filtering*. Thèse de Doctorat, LCAV, School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland, Octobre 2005.

[VG92] J. Vaisey and A. Gersho. Image compression with variable block size segmentation. *IEEE Transactions on Signal Processing*, 40(8) :2040–2060, Août 1992.

[VGP02] J. Viéron, C. Guillemot, and S. Pateux. Motion compensated 2D+t wavelet analysis for low rate fgs video compression. Dans *International Thyrrhenian workshop on digital communications*, Capri, Italy, Septembre 2002. invited paper.

[vHvdS98] J.H. van Hateren and A. van der Schaaf. Independent component filters of natural images compared with simple cells in the primary visual cortex. Dans *Proceedings of the Royal Society London B*, volume 265, pages 359–366, 1998.

[VKP00] V. Verma, D. Kao, and A. Pang. A flow-guided streamline seeding strategy. Dans *Proceedings of the 11th IEEE Visualization 2000 Conference*, Washington, DC, 2000. IEEE Computer Society.

[WA94] J.Y.A. Wang and E.H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5) :572–589, Septembre 1994.

[Wal91] G.K. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 34(4) :30–44, Avril 1991.

[Wan95] B. Wandell. *Foundations of Vision*. Sinauer Associates, 1995.

[Wat87] A.B. Watson. The cortex transform : rapide computation of simulated neural images. *Computer Vision, Graphics, and Image Processing*, 39(3) :311–327, Septembre 1987.

[WBL02] Z. Wang, A. C. Bovik, and L. Lu. Why is image quality assessment so difficult ? Dans *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 3313–3316, Orlando,FL, Mai 2002.

[WBSS04] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment : From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4) :600–612, Avril 2004.

[Wer38] M. Wertheimer. *Principles of perceptual organisation*. W. H. Ellis, London, 1938.

[Wie03] M. Wien. Adaptive deblocking filter. *IEEE Transaction on Circuits and Systems for Video Technology*, 13 :604–613, Juillet 2003.

[WJ01] H. Watanabe and K. Jinzenji. Sprite coding in object-based video coding standard : MPEG-4. Dans *Multiconference on Systemics, Cybernetics and Informatics*, volume 13, pages 420–425, Juillet 2001.

[WL94] Y. Wang and O. Lee. Active mesh - A feature seeking and tracking image sequence representation scheme. *IEEE Transactions on Image Processing*, 3(5) :610–624, Septembre 1994.

[WL96a] Y. Wang and O. Lee. Use of 2D deformable mesh structures for video coding. Part I — The synthesis problem : Mesh based function approximation and mapping. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(6) :636–646, Décembre 1996.

[WL96b] Y. Wang and O. Lee. Use of 2D deformable mesh structures for video coding. Part II — The analysis problem and a region-based coder employing an active mesh representation. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(6) :647–659, Décembre 1996.

[WLB04] Z. Wang, L. Lu, and A. Bovik. Video quality assessment based on structural distortion measurement. *Signal Processing : Image Communication, special issue on objective video quality metrics*, 19(2) :121–132, Février 2004.

[WNC87] I. H. Witten, R. M. Neal, and J. G. Cleary. Arithmetic coding for data compression. *Commun. ACM*, 30(6) :520–540, 1987.

[Wol94] George Wolberg. *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos, CA, 1994.

[WRCB02] M.B. Wakin, J.K. Romberg, H. Choi, and R.G. Baraniuk. Image Compression Using an Efficient Edge Cartoon + Texture Model. Dans *IEEE Data Compression Conference*, pages 43–52, Snowbird, Utah, Avril 2002.

[WSB03] Z. Wang, H. R. Sheikh, and A. C. Bovik. Objective video quality assessment. Dans B. Furht and O. Marques, editors, *The Handbook of Video Databases : Design and Applications*. CRC Press, 2003.

[WXCM99] A. Wang, Z. Xiong, P. A. Chou, and S. Mehrotra. Three-Dimensional Wavelet Coding of Video with Global Motion Compensation. Dans *Data Compression Conference*, pages 404–413, 1999.

[WZVS06] D. Wang, L. Zhang, A. Vincent, and F. Speranza. Curved Wavelet Transform for Image Coding. *IEEE Transactions on Image Processing*, 15(8) :2413–2421, Août 2006.

[XWX+04] R. Xiong, F. Wu, J. Xu, S. Li, , and Y-Q. Zhang. Barbell lifting wavelet transform for highly scalable video coding. Dans *Picture Coding Symposium*, San Francisco, CA, Décembre 2004.

[XXLZ00] J.-Z. Xu, Z. Xiong, S. Li, and Y.-Q. Zhang. Boundary effects in 3D wavelet video coding. Dans A. G. Tescher, editor, *SPIE Applications of Digital Image Processing XXIII*, volume 4115 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 406–417, Décembre 2000.

[XXLZ01] J.-Z. Xu, Z. Xiong, S. Li, and Y.-Q. Zhang. Three-dimensional embedded subband coding with optimized truncation (3D ESCOT). *Applied and Computational Harmonic Analysis*, 10 :290–315, 2001.

[XXWL07] R. Xiong, J. Xu, F. Wu, and S. Li. Barbell-lifting-based 3D wavelet coding scheme. *IEEE Transaction on Circuits and Systems for Video Technology*, 2007.

[Yar02] L. Yaroslavsky. Fast signal sinc-interpolation and its applications in signal and image processing. Dans *SPIE Image Processing : Algorithms and Systems*, volume 4667, San Jose, CA, Janvier 2002.

[YWCW05] F. Yang, S. Wan, Y. Chang, and H.R. Wu. A novel objective no-reference metric for digital video quality assessment. *IEEE Signal Processing Letters*, 12(10) :685–688, Octobre 2005.

[ZM00] S. Zhu and K.K. Ma. A new diamond search algorithm for fast block-matching motion estimation. *IEEE Transactions on Image Processing*, 9(2) :287–290, Février 2000.

[ZRMZ05] Y. Zhong, I. Richardson, A. Miller, and Y. Zhao. Perceptual quality of H.264/AVC deblocking filter. Dans *IEE International Conference on Visual Information Engineering*, pages 379–384, Glasgow, Avril 2005.

Table des figures

Abstract

Limits of standard separable wavelets are well known in the 2D case. Their fixed rectangular support cannot capture the regularity along curved contours and hence they fail to accurately represent the geometry in images. This results in a large number of non zero coefficients in the wavelet domain and produces a ringing artefact near contours when approximating the signal with a small number of coefficients. To improve the wavelet representation, second generation wavelet have been built. The most common approach is to warp the wavelet to adapt it to the geometrical content in an image.

In this thesis, we address the adaptivity issue in a different fashion. The idea is to warp the image content in order to adapt it to the standard separable wavelet. The warping is represented by an active 2D mesh. The adaptation criterion is the description cost of the warped image. An energy minimization is described to compute the parameters of the mesh. This optimization is similar to a motion estimation between two frames. After this analysis step, the image is represented by a warped image with smaller coding cost, and a set of warping parameters. After encoding and decoding the information, the original image can be synthesized by inverting the warping. Our spatial analysis-synthesis scheme is compared to JPEG2000 in terms of coding efficiency. Visually, a better reconstruction of contours is noticed with a significant reduction of the ringing artefact.

Keeping the same idea to adapt the content of images to a fixed decomposition filter, we then propose a spatio-temporal analysis-synthesis scheme dedicated to videos. The analysis takes a group of frames (GOF) as input and outputs a group of warped frames which content is adapted to a fixed horizontal-vertical-temporal 3D decomposition. The scheme is designed so that only one geometry must be estimated and transmitted for each GOF. Compression results are presented. They were obtained by using active meshes to represent both the geometry and the motion. Although only one geometry must be encoded, we show that its coding cost remains too important to produce a significant visual improvement compared to an analysis-synthesis scheme which only takes the motion into account.

Key words

Still image coding, video coding, analysis-synthesis, wavelets, active mesh, motion, geometry, scalability

Résumé

Les limites de l'ondelette séparable standard, dans le cas 2D, sont bien connues. Le support rectangulaire fixe de l'ondelette ne permet pas d'exploiter la géométrie des images et en particulier les corrélations le long de contours courbes. Ceci se traduit par une dispersion de l'énergie des coefficients dans le domaine ondelette et produit une phénomène de rebonds gênant visuellement lors d'une approximation avec un petit nombre de coefficients. Pour y remédier, une seconde génération d'ondelettes est née. L'approche la plus courante est de déformer le noyau d'ondelette pour l'adapter au contenu géométrique d'une image.

Dans cette thèse, nous proposons d'aborder le problème d'adaptativité sous un angle différent. L'idée est de déformer le contenu d'une image pour l'adapter au noyau d'ondelette séparable standard. La déformation est modélisée par un maillage déformable et le critère d'adaptation utilisé est le coût de description de l'image déformée. Une minimisation énergétique similaire à une estimation de mouvement est mise en place pour calculer les paramètres du maillage. A l'issue de cette phase d'analyse, l'image est représentée par une image déformée de moindre coût de codage et par les paramètres de déformation. Après codage, transmission et décodage de ces informations, l'image d'origine peut être synthétisée en inversant la déformation. Les performances en compression de ce schéma par analyse-synthèse spatiales sont étudiées et comparées à celles de JPEG2000. Visuellement, on observe une meilleure reconstruction des contours des images avec une atténuation significative de l'effet rebond.

Conservant l'idée d'adapter le contenu des images à un noyau de décomposition fixe, nous proposons ensuite un schéma de codage par analyse-synthèse spatio-temporelles dédié à la vidéo. L'analyse prend en entrée un groupe d'images (GOF) et génère en sortie un groupe d'images déformées dont le contenu est adapté à une décomposition 3D horizontale-verticale-temporelle fixe. Le schéma est conçu de sorte qu'une seule géométrie soit estimée et transmise pour l'ensemble du GOF. Des résultats de compression sont présentés en utilisant le maillage déformable pour modéliser la géométrie et le mouvement. Bien qu'une seule géométrie soit encodée, nous montrons que son coût est trop important pour permettre une amélioration significative de la qualité visuelle par rapport à un schéma par analyse-synthèse exploitant uniquement le mouvement.

Mots clés

Codage d'images fixes, codage vidéo, analyse-synthèse, ondelettes, maillage déformable, mouvement, géométrie, scalabilité